Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 516–523

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 23 - 26, 2021, Warsaw, Poland

A Hybrid Data-driven Model for Intrusion Detection in VANET

Hind Bangui*, Mouzhi Ge, Barbora Buhnova

[a]*Faculty of Informatics, Masaryk University, Brno, 602 00, Czech Republic*

Abstract

Nowadays, VANET (Vehicular Ad-hoc NETwork) has gained increasing attention from many researchers with its various applications, such as enhancing traffic safety by collecting and disseminating traffic event information. This increased interest in VANET has necessitated greater scrutiny of machine learning (ML) methods used for improving the security capabilities of intrusion detection systems (IDSs), such as the need to solve computationally intensive ML problems due to the increased vehicular data. Therefore, in this paper, we propose a hybrid ML model to enhance the performance of IDSs by dealing with the explosive growth in computing power and the need for detecting malicious incidents timely. The proposed approach mainly uses the advantages of Random Forest to detect known network intrusions. Besides, there is a post-detection phase to detect possible novel intruders by using the advantages of coresets and clustering algorithms. Our approach is evaluated over a very recent IDS dataset named CICIDS2017. The preliminary results show that the proposed hybrid model can increase the utility of IDSs.

Keywords: VANET, Clustering, IDS, Coreset, Security , Data Approximation;

1. Introduction

VANET (Vehicular Ad-hoc Network) [1, 2] is emerging as a promising communication technology in intelligent transportation systems. It aims to support safety-critical traffic applications by enabling vehicles to exchange information among themselves to avoid critical situations through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, such as avoiding traffic jams, black ice, and unseen obstacles. Furthermore, VANET provides real-time services to drivers, such as Internet access on the move, weather information, and route recommendation. This particular type of mobile ad-hoc network not only uses V2V and V2I to disseminate data but also it uses the potential of various communications, namely V2X (Vehicle-to-Everything) [3], to improve the performance of mes-

* Corresponding author.
E-mail address: hind.bangui@mail.muni.cz

10.1016/j.procs.2021.03.065

Table 1. Machine learning algorithms in VANET

Papers	Methods used	Description of the solution
[7]	SVM	Develop a trust-aware framework to improve the safety of vehicles.
[8]	k-means based on self-organized map	Address the high accuracy and precision demands of intrusion detection systems.
[9]	Neural Networks	Propose a model for estimation traffic and anomaly detection.
[10]	SVM	Use an optimized IDS for securing a multi-cluster head anomaly.
[11]	SVM, Neural network	Propose an intelligent IDS for dealing with grey hole and rushing attacks in self-driving vehicular networks.
[12]	SVM	Develop a collaborative intrusion detection framework for getting a low false positive rate, high detection rate, lower communication overhead, and faster attack detection.
[13]	K-means, linear splines	Propose an IDS model to deal with the nature of VANET.
[14]	Random Forest, k-NN	Propose an IDS model against spoofing attacks in connected electric vehicles.
[15]	Naive Bayes, Logistic Regression	Propose a distributed IDS for detecting unknown intrusions in VANET.
[16]	Bayesian Filter	Propose an IDS for identifying attackers and reducing false positive of nodes.
[17]	Neuro-Fuzzy	Propose an enhanced framework for attack detection in VANET.
[18]	Neural Networks	Propose a hybrid intrusion detection for connected self-driving vehicles.

sage dissemination in VANET, and allow vehicles to communicate with other entities over multiple communications (e.g., Vehicle-to-Pedestrian)[3]. As a result, VANET has attracted the attention of research and industrial communities thanks to its gradual maturity in the transportation domain.

Due to the high data dissemination through V2X communications, VANET technology is one of the smart technologies that play a crucial role in critical infrastructure systems that are vital for the national health, security, and economy [4, 5]. For example, the smart grids infrastructure requires transmitting a massive amount of real-time data to computing centers through communication networks to enable efficient system operation, such as detecting anomalies and failures, forecasting electricity usage, and evaluating power quality. In such a situation, the exploitation of VANET as a temporal computing core could capture, manage, and process the mounting volume of data timely. As a result, it can make the smart grids infrastructure more reliable since VANET supports real-time processing requirements. However, network security issues still significantly affect VANET and its fusion with critical systems that need to store, transmit, archive, and retrieve data via networks timely [6]. Yet, the development of data-driven intrusion detection systems (IDSs) is required in VANET to assure its advancement and avoid attacks that might cause destructive consequences.

In this paper, we propose a hybrid data-driven detection model based on a machine learning with two main components: the first detection component aims to detect the known attacks by using a classification algorithm. As the second component, the anomaly detection is to filter the dishonest nodes by using a clustering algorithm based on the coresets technique. The model is used to improve the security in VANET by enhancing the accuracy of detection against various attacks in real-time.

The remainder of the paper is organized as follows. Section 2 presents a brief review of the machine learning application for the detection against attacks. Section 3 describes the proposed hybrid detection based on machine learning techniques. Section 4 describes the experimental environment, along with the discussion of experiment results. Section 5 provides concluding remarks that could open new research directions for security in VANET. Finally, Section 6 summarizes the paper and outlines future work.

2. Related Work

Different machine learning algorithms have been applied for IDSs in VANET to detect various attacks simultaneously (Table 1), such as in [8], an intelligent scheme using clustered self-organized map (SOM) has been presented in order to address the high accuracy and precision demands of intrusion detection systems. To do that, k-means is applied to SOM neurons during the detection phase. Likewise, in [7], the authors have proposed a novel trust-aware approach for both intrusion detection and prevention in VANET by combing modified promiscuous mode for data collection and SVM for data analysis to establish a precise trust score table for each vehicle. As a result, this method can improve the safety of vehicles by detecting any sign of a compromised node that can affect the network performance.

Despite the large implication of machine-learning based IDSs algorithms (Table 1), IDSs are still facing many security issues (e.g., high false-positive rate and false-negative rate) due to increased vehicular data that is not well addressed. Indeed, the existing work only focuses on using machine learning methods for analyzing small VANET databases without suggesting any strategy to deal with large databases and speed up the process of data analytics (e.g., training time) to go with the needs of volatilized and distributed VANET environments. Moreover, applying machine-learning algorithms on very big traffic datasets containing noise representatives might reduce the attack detection accuracy. Yet, the traditional implication of machine learning for IDSs has limitations in scaling up to meet the security VANET requirements.

3. Data-driven Intrusion Detection Model

We propose a hybrid detection model that is based on data classification and clustering with coresets. The process of this data-driven model is described in Figure 1. It consists of two main components that are further described below.

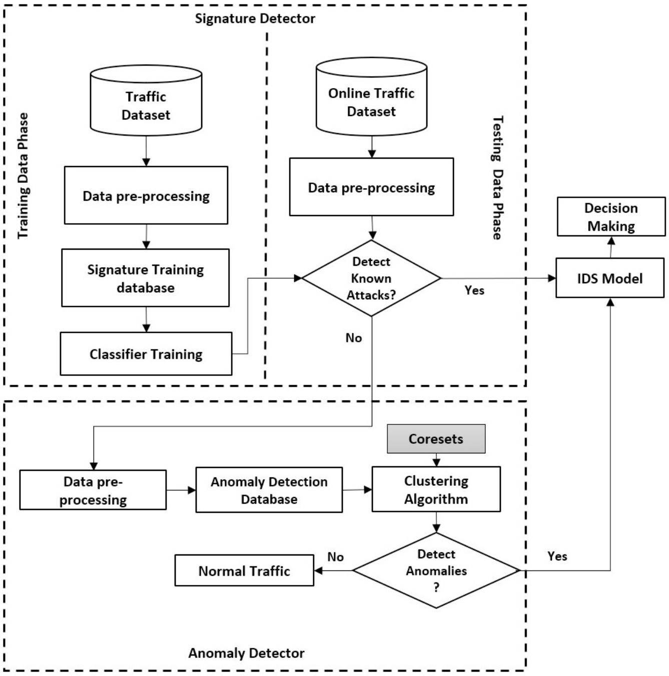

Fig. 1. Hybrid data-driven model for intrusion detection

3.1. Signature Detector

The first component of the proposed model is for detecting known attacks by using a classification algorithm. In other words, this component is dedicated to the signature detection technique that refers to the detection of known attacks by using an existing database behavior with specific attacks, which are already registered. As a result, this model helps to recognize malicious attacks faster by using a classifier for matching their signatures against those stored in the signature database.

We propose to use Random Forest as a classifier due to its advantages compared to other supervised algorithms, such as dealing with the imbalanced database easily, randomizing the selection of feature variables to improve the decision tree method, and being faster to predict the class label [19, 20]. Random Forest is an ensemble technique [21], in which “individual weak learners” collaborate to form a “strong learner.” Random Forest algorithm generates many classification trees to avoid overfitting to training datasets, where each tree is constructed by using a bootstrap randomized re-sampling technique [22]. For making a classification decision, the algorithm obtains the prediction results from each tree, establishes a voting mechanism, and finally takes a plurality vote among the classifiers.

3.2. *Anomaly Detector*

Since the signature detection technique cannot recognize new intrusions that are not stored in the labeled misuse database, the second component of our proposed model is dedicated to anomaly detection. This latter refers to the detection of unknown intrusions by using an analyzer model that is capable of learning and detecting the event deviating from normal behavior as an intrusion behavior. Therefore, we use clustering algorithms to detect novel intrusions, particularly, we use weighted k-means [23, 24] since we exploit the data labeling from the first component.

Algorithm 1 Weighted K-means

Require: Data set $D = \{x_1, x_2, \ldots, x_n\}$, K for clustering step.
Ensure: the set of k clusters $S = \{S_1, S_2, S_3, ..., S_{k-1}\}$.

```
Use the feature importance that is calculated by Random Forest as the weights vector W.
for each $x_i$ do
    $\omega_{x_i} = RandomForest(x_i)$.
end for
Sort Features $x_i$ to obtain its importance set. $W = Sort(\omega_{xi})$
Select $K$ objects as the initial centroids.
Repeat
Assign each object to its nearest centroid $C_i$
Find the distance between the centroids using the weighed euclidean distance.
Re-compute the centroids $C_i$ of all clusters.
Until there is no reassignment of objects to new centroids
```

On the other hand, we apply the coreset method to deal with computational complexity in clustering [25] and the large volume of vehicular datasets by extracting critical contents without examining all data content, and then enable IDSs to ensure timely network security. This approximate technique is not only just giving a quick viewability of original data, but also helps with scaling Big Data Analytics techniques during data processing [25]. The details of the implication of coresets for clustering are given below.

3.2.1. *Construction of Coresets via Clustering*

We exploit the results of Random Forest algorithm from the Signature Detector section to assign each data point to each cluster with a certain probability. Algorithm 2 represents the details followed to construct coreset C and approximate clustering. It is composed of three steps described below.

In the first step, we use a unified framework for coreset construction described in [26]. This generic solution aims to obtain a rough approximation B of the set D by labeling the original data points as sampled or removed. Then, we get the initial clustering that will be used to select the set of k centroids in step 2 and sample the data points containing coreset C in step 3.

The second step consists of adding the other k-1 clusters by projecting samples according to a probability onto the bicriteria centers B. Moreover, in this work, the probability is computed by using the feature importance measurements of Random Forest as the weights vector. Then, we construct clusters P by applying weighed k-means (Algorithm 1) over the representatives of P.

In the last step, we apply a coreset construction introduced by [27] to sample the points by calculating new weights, which are associated with the sampling probabilities. Finally, we can build a proper coreset from the entire dataset based on the adapted weights; simultaneously, the representatives of P are updated.

3.2.2. *Post-Coresets processing*

After constructing approximate clusters, we referred to our previous work [28] in which we proposed a post-Coresets processing to find sample points that have low representativeness and remove them from the sample. The details of the method are given in Algorithm 3.

Given a dataset $P = \{x_i\}$, for $i = 1, 2, \ldots, n$ and a given $\epsilon > 0$, the method first calls Algorithm 4 to obtain $C = \{y_j\}$ and $P(y_j)$ for $j = 1, 2, \ldots, c$. Line 4 determines which sample points will be selected into our sample C'. This is

Algorithm 2 Coreset Construction via Weighted k-means

Require: Data set $D = \{x_1, x_2, \ldots, x_n\}$, number of clusters K, failure probability δ, approximation factorα, coreset size m.

```
Step I: Obtaining the initial approximate partition B of the clustering solution
$D\prime \leftarrow D$; $B \leftarrow \emptyset$; $b \leftarrow 10dkln(1/\delta)$;
while $| D\prime |> b$ do
    $S \leftarrow$ Sample $b$ points uniformly at random from $D\prime$;
    $P \leftarrow [| D\prime | /2]$ points from $D\prime$ closest to $S$;
    $D\prime \leftarrow$ Remove $D\prime/P$ points $x \in D\prime$ closest to $S$ from $D\prime$;
    Set $B \leftarrow B \bigcup S$;
end while
Set $B \leftarrow B \bigcup D\prime$;
Step II: Adding K-1 Cluster centers
Substitute the weights vector ($W_x$) by the feature importance measurements of Random Forest, $W_x \leftarrow RandomForest$(x);
for $j \leftarrow 2, ..., K$ do
    for $x \in D$ do
        $Pr(x) \leftarrow \frac{W_x d(x,B)^2}{\sum_{x\prime \in D} W_{x\prime} d(x\prime,B)^2}$;
    end for
    Sample $x \in D$ with probability Pr(x);
    $B \leftarrow B \bigcup \{x\}$;
end for
Induce dataset partition $P$,$P \leftarrow B(D)$, where $B(D) = \{Blocks(D)\}_{Blocks \in B}$ and $Blocks(D) = \{x \in D : x$ lies on $Blocks \in B\}$;
Step III: Constructing Coreset
for $c \in B$ do
    $D_c \leftarrow$ point $x \in D$ whose closest center in $B$ is $c$. Ties broken arbitrarily;
end for
for $c \in B$ and $x \in D_c$ do
    $s(x) \leftarrow \alpha d(x,B)^2 + \frac{2\alpha}{D_c} \sum_{x\prime \in Dc} d(x\prime,B)^2 + \frac{2}{D_c} \sum_{x\prime \in D} d(x\prime,B)^2$
end for
for $x \in D$ do
    $q(x) \leftarrow \frac{s(x)}{\sum_{x\prime \in D} s(x\prime)}$;
end for
$C \leftarrow$ Sample $m$ weighted points from $D$, where each point $x$ is sampled with probability $q(x)$ and assigned a weight $\frac{1}{m.q(x)}$;
Update dataset partition $P$, $P \leftarrow B(C)$;
Return $C$ and $P$;
```

Algorithm 3 Post-Coresets-processing

Require: $P = \{x_i\}$, for $i = 1, 2, \ldots, n$, a tolerance $\epsilon > 0$.

Ensure: A sample $C = \{y_j\}$ and $P(y_j)$, for $j = 1, 2, \ldots, c$.

```
Call Algorithm 2 for $P$ and $\epsilon$ to obtain $C = \{y_j\}$ and $P(y_j)$.
$C' = \emptyset$.
for all $y_j \in S$ do
    if $|P(y_j)|$ is greater than a threshold then
        $y_k^* =_{y_k \in P(y_j)} \sum_{y_l \in P(y_j)} d(y_k, y_l)$.
        $C' = C' \cup \{y_k^*\}$.
    end if
end for
$C = C'$.
return $C$ and $P(y_j^*)$, for $j = 1, 2, \ldots, c'$, where $c' \leq c$.
```

Table 2. Number of records

Type	Label	Number of instances
Normal	BENIGN	233 107
DoS/DDoS Attacks	DDoS, DoS, DoS slowloris, DoS Slow,Httptest, DoS Hulk,DoS GoldenEye,Heartbleed	24 350
Botnet Attack	Bot	1560
PortScan Attack	PortScan	8000
Brute Force Attack	FTP-Patator, SSH-Patator	4000
Infiltration Attack	Infiltration	30
Web Attacks	Web Attack-Brute Force, Web Attack-XSS, Web Attack-Sql Injection	2050
Toral of Instances	273 097	

performed using a threshold. $|P(y_j)|$ denotes the number of patterns in P with $y_j \in C$ being their representative. Small value of $|P(y_j)|$ means that the representativeness of y_j is low. Accordingly, it is removed from the sample. The value of the threshold should be chosen based on the distribution characteristics of datasets. For $y_j \in C$ that is not removed, line 5 computes the center of the group represented by y_j, to consider replacing it. The center here, denoted with y_k^*, is defined to be the point in $P(y_j)$ such that the total distance to all others in the group is minimized. The set C' including such y_k^* is the output sample of the algorithm.

4. Evaluation

We evaluated the proposed model in MATLAB environment by conducting experiments on the IDS dataset of CICIDS2017 [29]. This real-world traffic dataset covers the most cutting-edge frequent attack scenarios based on simulation of seven attack families, namely: brute force attack, heart-bleed attack, botnet, DoS attack, DDoS attack, web attack, and infiltration attack. A total number of 80 features were extracted based on the information present in the PCAP file. The total number of records used in this experiment is 273 097. Table 2 shows the detail of records used in this evaluation, where the dataset is divided into two parts using train-test_split [30], 80% for training and 20% for testing the model.

We have compared our proposed model with other machine learning algorithms, particularly, the work including coresets. Indeed, we found one work that has combined coresets with Bayesian machine learning for network intrusion detection systems [31]. The performance of IDSs is evaluated in term of F1-score, accuracy, and computation time.

4.1. Experimental Results

Figures 2 and 3 present the evaluation metrics, namely, F1-score, and Accuracy. Figure 3 shows that our proposed method has achieved the highest of 96.93% Accuracy while the lowest is with Bayesian-Coresets 82.4%. The F-measure of the proposed method (Figure 2) obtained 94.41% compared to the existing classification models, such as Bayesian-Coresets and SVM attained 77.02% and 82.01% respectively. This is because our model is composed of two phases, where the first one corresponds to the misuse detection approach in IDS, and the second phase corresponds to the anomaly detection approach that detects the compromise indicators. On the other hand, we noticed that CNN reached significant results (Figures 2 and 3), i.e. it is very close to the highest accuracy with 95.14%; however, our proposed model's computation time (140.6s) was much lower than CNN (754.18s) thanks to the use of random forest and coresets models for near-real time (Figure 4). As a result, the approximate technique aids the hybrid learning approach to produce better results for IDS models.

5. Discussions

Data approximation using coresets [32] has gained significant attention in Big Data due to the ability of coresets to produce summaries, which are compact representatives of the original data with a provable approximation guarantee. Data approximation is performed on data by applying several rules enabling learning methods to use only the necessary data. It prevents data replication by reducing the size of the collected data, and removes all ineffective data. As a result, it supports the development of IDSs by predicting future sequences in fast manner, providing early warning features

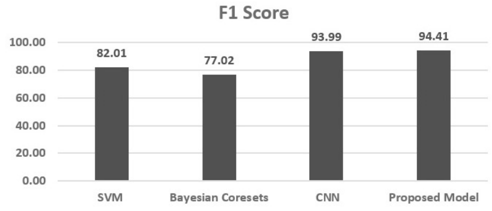

Fig. 2. F1-Score (%)

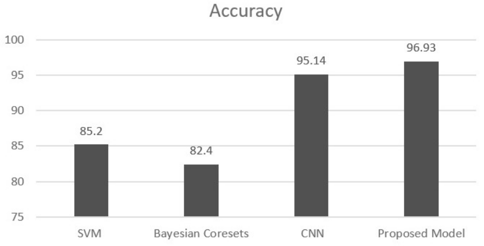

Fig. 3. Accuracy (%)

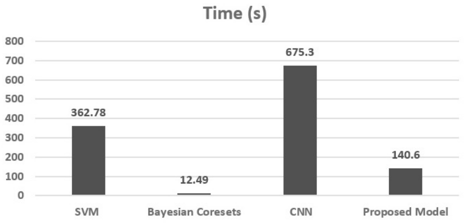

Fig. 4. Time (s)

to different types of intrusion, identifying the places where it is vulnerable to an attack, capturing malicious events before they occur, improving estimated risk measures of security alerts.

Furthermore, data approximation can become one of the key steps to filter the expected attack automatically and support to connect different critical information services in VANET that benefit from the transmitted information being secured. However, to achieve a faster IDS, it is important to understand how data augmentation influences the construction of summarized data, what type of approximation techniques should be used for machine learning and how data approximation can affect the IDS decision making.

6. Conclusion

In this paper, we have proposed a hybrid machine learning model to efficiently conduct comprehensive intrusion detection in VANET. The proposed solution combines data classification and coresets-based clustering. It takes advantage of coresets to achieve lower overhead of computational time consumption and enhances the reasoning process of IDSs in VANET. To validate the proposed model, we have conducted an experiment with a public IDS evaluation dataset, CICIDS2017. Our preliminary results have shown that the proposed solution can increase the detection accuracy, which improves further the utility of IDSs. As future work, we plan to deploy our solution in different real-world scenarios such as Internet of Vehicles and benchmark the performance of the proposed solution.

Acknowledgment

The work was supported from ERDF/ESF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] Nan Cheng, Feng Lyu, Jiayin Chen, Wenchao Xu, Haibo Zhou, Shan Zhang, and Xuemin Sherman Shen. Big data driven vehicular networks. *IEEE Network*, 32(6):160–167, 2018.

[2] Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455, 2017.

[3] Chang An and Celimuge Wu. Traffic big data assisted v2x communications toward smart transportation. *Wireless Networks*, 26(3):1601–1610, 2020.

[4] Mouzhi Ge, Hind Bangui, and Barbora Buhnova. Big data for internet of things: A survey. *Future Generation Computer Systems*, 87:601–614, 2018.

[5] Hind Bangui, Said Rakrak, Said Raghay, and Barbora Buhnova. Moving to the edge-cloud-of-things: recent advances and future research directions. *Electronics*, 7(11):309, 2018.

[6] Rasheed Hussain, Jooyoung Lee, and Sherali Zeadally. Trust in vanet: A survey of current solutions and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] Erfan A Shams, Ahmet Rizaner, and Ali Hakan Ulusoy. Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks. *Computers & Security*, 78:245–254, 2018.

[8] Muder Almi'ani, Alia Abu Ghazleh, Amer Al-Rahayfeh, and Abdul Razaque. Intelligent intrusion detection system using clustered self organized map. In *2018 Fifth International Conference on Software Defined Systems (SDS)*, pages 138–144. IEEE, 2018.

[9] Laisen Nie, Yongkang Li, and Xiangjie Kong. Spatio-temporal network traffic estimation and anomaly detection based on convolutional neural network in vehicular ad-hoc networks. *IEEE Access*, 6:40168–40176, 2018.

[10] Sparsh Sharma and Ajay Kaul. Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized ids for vanet. *Vehicular Communications*, 12:23–38, 2018.

[11] Khattab M Ali Alheeti, Anna Gruebler, and Klaus McDonald-Maier. Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks. *Computers*, 5(3):16, 2016.

[12] Hichem Sedjelmaci and Sidi Mohammed Senouci. An accurate and efficient collaborative intrusion detection framework to secure vehicular networks. *Computers & Electrical Engineering*, 43:33–47, 2015.

[13] David A Schmidt, Mohammad S Khan, and Brian T Bennett. Spline-based intrusion detection for vanet utilizing knot flow classification. *Internet Technology Letters*, page e155, 2020.

[14] Dimitrios Kosmanos, Apostolos Pappas, Leandros Maglaras, Sotiris Moschoyiannis, Francisco J Aparicio-Navarro, Antonios Argyriou, and Helge Janicke. A novel intrusion detection system against spoofing attacks in connected electric vehicles. *Array*, 5:100013, 2020.

[15] Xiaoyun Liu, Gongjun Yan, Danda B Rawat, and Shugang Deng. Data mining intrusion detection in vehicular ad hoc network. *IEICE TRANSACTIONS on Information and Systems*, 97(7):1719–1726, 2014.

[16] Jay Rupareliya, Sunil Vithlani, and Chirag Gohel. Securing vanet by preventing attacker node using watchdog and bayesian network theory. *Procedia computer science*, 79:649–656, 2016.

[17] Jasleen Kaur, Tejpreet Singh, and Kamlesh Lakhwani. An enhanced approach for attack detection in vanets using adaptive neuro-fuzzy system. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pages 191–197. IEEE, 2019.

[18] Khattab M Ali Alheeti and Klaus McDonald-Maier. Hybrid intrusion detection in connected self-driving vehicles. In *2016 22nd International Conference on Automation and Computing (ICAC)*, pages 456–461. IEEE, 2016.

[19] C Sothe, CM De Almeida, MB Schimalski, LEC La Rosa, JDB Castro, RQ Feitosa, M Dalponte, CL Lima, V Liesenberg, GT Miyoshi, et al. Comparative performance of convolutional neural network, weighted and conventional support vector machine and random forest for classifying tree species using hyperspectral and photogrammetric data. *GIScience & Remote Sensing*, 57(3):369–394, 2020.

[20] Ting Lan, Hui Hu, Chunhua Jiang, Guobin Yang, and Zhengyu Zhao. A comparative study of decision tree, random forest, and convolutional neural network for spread-f identification. *Advances in Space Research*, 2020.

[21] Anjaneyulu Babu Shaik and Sujatha Srinivasan. A brief survey on random forest ensembles in classification model. In *International Conference on Innovative Computing and Communications*, pages 253–260. Springer, 2019.

[22] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[23] Renato Cordeiro De Amorim and Boris Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, 45(3):1061–1075, 2012.

[24] Hind Bangui, Mouzhi Ge, and Barbora Buhnova. A research roadmap of big data clustering algorithms for future internet of things. *International Journal of Organizational and Collective Intelligence (IJOCI)*, 9(2):16–30, 2019.

[25] Dan Feldman. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 23–44. Springer, 2020.

[26] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

[27] Mario Lucic, Olivier Bachem, and Andreas Krause. Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures. In *Artificial intelligence and statistics*, pages 1–9, 2016.

[28] Le Hong Trang, Hind Bangui, Mouzhi Ge, Barbora Buhnova, et al. Scaling big data applications in smart city with coresets. 2019.

[29] *University of New Brunswick, Canadian Institute for Cybersecurity: Intrusion Detection Evaluation Dataset (CICIDS 2017)*, Accessed 15 September 2020.

[30] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[31] Fabio Massimo Zennaro. Analyzing and storing network intrusion detection data using bayesian coresets: A preliminary study in offline and streaming settings. *arXiv preprint arXiv:1906.08528*, 2019.

[32] Alexander Munteanu and Chris Schwiegelshohn. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *KI-Künstliche Intelligenz*, 32(1):37–53, 2018.