SUPPORT VECTOR MACHINE BASED INTRUSION DETECTION SYSTEM IN FOG COMPUTING

M. Ramkumar
Department of Computer Science and Engineering, Gnanamani College of Technology, India

Abstract

In this paper, SVM intrusion detection classification is used to detect the intrusions in fog based mobile edge computing (MEC). However, chosen characteristics such as a mean, limit and median system and the suggested detection scheme can be used regardless of the type of propagation such that the traffic strength to the node is affected by an increase or decrease in intrusion or attack. The SVM is preferred for its efficient efficiency among other machine learning algorithms. The similarity of output in the experimental results section between the various algorithms supports our point. In addition, the experimental findings demonstrate the light weight of the algorithm. In addition, we provided a comparative comparison with other machine-driven classifiers of the SVM based classifier to demonstrate how accurate SVM is than other techniques. We have submitted a comparison of the algorithm suggested with another IDS in literature for further verification. The findings indicate that an SVM-based IDS can be used to detect attacks satisfactorily.

Keywords:

Intrusion Detection, Fog Computing, Support Vector Machine, Parameters

1. INTRODUCTION

We embrace the age of the Internet of Things with the fast growth of intelligent devices (IoT). IoT systems include support for accessibility, geodistribution, knowledge of the location and low latency. However, these conditions cannot be fulfilled by cloud computing. The above-mentioned challenges are suggested to be solved in terms of edge paradigms such as fog computing (FC) and mobile edge computing (MEC). The nodes that are capable of performing computational activities in the FC and MEC hosts are called fog nodes/MEC hosts that can provide low-latency services. FC and MEC vary somewhat.

For example, mobile service providers usually deploy MEC hosts [4], and fog nodes consist of edge servers or communications and processing power equipment. However, they are closer to their network model and several features [5]. Cloud computing is extended to the edge of both. In this analysis, we are studying a generally used FC and MEC network model. The infrastructure of FC or MEC is generally composed of three layers: the cloud server; the fog node/MEC host layer; and the user device layer on which Fog nodes/MEC hosts are more user-specific and FC-specific computer nodes [6]. The aim of the fog nodes/MEC hosts is to provide service to network users with less latency, greater flexibility in connectivity and safer network communication [7].

MEC poses various network reliability and consistency problems as a modern network model. The terminal connected with the MEC node/MEC host may be, for example, a smart home appliance, a smart phone, an unmanned aircraft (UAV) or a VR device [8]. The terminal devices are often restricted by resources on the MEC system. Different network features such as denial of service (DoS), man in the center (MIM), rogue portal, privacy leakage, and service abuse could cause threats [5]. Since the FC network is attacked extensively, the advantages of FC are reduced by malicious attacks if there is no adequate safety or privacy defense.

We concentrate on security measures in order to efficiently manage the risks in the FC infrastructure and mitigate the resulting risk. The intrusion sensing device is one way (IDS). An IDS is a significant safety shield that rapidly recognizes network intrusion and security risks [9]. One of the most significant elements of IDS is the detection algorithm [10]. A vital infrastructure intrusion detection algorithm can detect intrusions correctly and promptly. The current research focuses therefore on the intrusion detection system to guarantee safety and address new problems to conform to the new FC model.

The first analysis of the safety issues in MEC is this article. Under the resources-constrained features of fog nodes/MEC hosts, a lightweight algorithm is suggested for intrusion detection. The computational resources of fog nodes/MEC hosts are fully used in this design. It deploys intrusion detection classifiers on fog nodes and MEC hosts and stores cloud server training data sets.

This paper proposes a general intrusion detection method in the MEC context according to the network characteristics of fog. This paper adds the sample collection procedure in the training phase on the basis of standard sample selection. This architecture enhances the classification algorithm to become lighter as fog nodes/MEC hosts perform tasks.

2. RELATED WORKS

Existing literature on MEC safety focuses primarily on the analysis of safety risks in MEC and the appropriate countermeasures [3] [6]-[9].

Intrusion detection studies in MEC [6] are available. Edge computing safety has been previously examined [2] and the author has indicated that MEC is an edge computing paradigm. This paper addressed in depth numerous safety problems in edge computing and security mechanisms. The Author noted that the security needs of both local fog nodes/MEC hosts and the whole network could be met with a fog IDS, which could delete permanent threats and provide an autonomous intrusion prevention mechanism. It was noted that an IDS is available for monitoring server-side intrusions or network-side attacks on the fog node/MEC host side or the Fog network/MEC network side (network-side detection) [1].

A new cloud-based mesh protection architecture has been proposed [6], which protects the mobile cloud network. This architecture. When cloudlet servers are viewed like fog

nodes/MEC hosts, the architecture is a host IDS. MEC is a cloud computer extension paradigm. Therefore, in this analysis we apply to the VM IDS in cloud computing.

For example, an intrusive detection system was implemented that was used in a cloud VM to capture and detect abusive actions of VMs based on the Smith-Waterman algorithm [7]. It enhanced the detection performance of the device and thus decreased the detection time to a certain degree.

In an earlier report [8], threats in federated cloud systems were analyzed and an abuse cataloging monitoring scheme was proposed. There was an approach called VAED used in VMs. Its findings appear promising to detect malware attacks based on evasion. Many experiments detectable intruders. Since these studies are conducted in cloud, however, there is no emphasis on IDS in resource-restricted settings, in particular MEC. In other words, lightweight IDS in MEC deserve to be studied. Techniques such as statistical, data mining detection, expert-system-based detection, support-based vector-machine (SVM) detection, genetically algorithm-based detection, and nervous network-based detection provide intrusion detection.

3. INTRUSION DETECTION - BACKGROUND

Fog networks and MEC networks provide consumer computers, fog nodes and MEC hosts, as well as cloud centres. The standard cloud computing model is typically different. A cloud processing center maintains several fog nodes/MEC hosts and operates them. Fog nodes/MEC hosts situated between the network center and the customer are computers powerfully located on the edge of the network. Fog nodes/MEC hosts can manage computer activities and provide user devices with network resources. User devices are heterogeneous, including smart, UAV and other interconnected devices. In a fog network/MEC network, fog nodes/MEC hosts and terminals are assumed to be likely to be targeted and therefore not secure. Taking into account the network fog/MEC structure, we present the network attack intrusion diagram in MEC as follows.

MEC layer network connection support for consumer devices provides fog nodes/MEC hosts/MEK hosts and the sharing of data with the cloud storage server as well. Threats to the security of the fog nodes/MEC hosts/MEC hosts and user devices are caused directly or indirectly. Network nodes in MEC are widespread and have little effective physical defense. They are vulnerable to hostile aggressor invasion. Traditional IDS have been unable to fulfill fog detection needs through its detection capacity and response time. Consequently, it was a valuable research guideline for MEC data protection to create an effective intrusion detection system in the MEC area.

The intrusion detection mechanism in a fog network/MEC network is the process in which the IDS decides if the host state or network link data is legal through a security audit. In addition, the implementation of the fog network/MEC network IDS is linked to the safety of the whole system. The Cloud computing center will calculate and store data on all fog nodes/MEC servers/MEC hosts. We opted to use IDS in the fog network/MEC network with the help of cloud storage capabilities and fog nodes/MEC hosts with a restricted calculation capacity to spread IDS computational load fairly. The implementation scheme has certain explanations. In terms of sensing reliability, the network connectivity cost will increase if an IDS is only implemented in a cloud processing center, as all data must be transmitted and processed by the cloud server.

In addition, the cloud server's calculation load would also increase. In security terms, the cloud center can quickly become the victim of hacks, as the primary node of the whole network. When the central node is targeted, it is impossible to trust the data collected from the fog node/MEC host detectors. Data sets for intrusion detection are required if an intrusion detection neural network is used. The training data is normally high, so the training time is increased significantly if only the cloud computing center performs the training, thereby compromising the quality of the training.

The various fog nodes/MEC hosts will result in quite different network environments in MEC thanks to the heterogeneity of user devices. Such fog nodes/MEC hosts, for example, mostly provide cars with networking facilities and therefore connect with vehicle sensors or car control systems, while other fog nodes/MEC hosts only support smartphones. In addition, the fog node/MEC host consumer demographic often dynamically changes: a user device may at any time enter or leave a fog node/MEC host.

3.1 INTRUSION DETECTION SCHEME

We suggest an overall architecture for MEC intrusion detection systems in order to respond to the complex fog network/MEC network and secure the protection and high performance of the fog IDS. This schema benefits from fog nodes/MEC hosts and cloud servers' processing capacities and space for storage. The architecture consists of the encoding, identification and exploitation of IDS information. According to fog network/MEC data stream, the scheme is divided into six layers.

- User Equipment Layer: MEC consumer equipment is heterogeneous, including personal computers (PCs), smart terminals, IOVs and sensors. The devices can use various protocols to reach different fog nodes/MEC hosts.
- Network Layer: It offers connection services for various network fog/MEC protocols. It is responsible for the reception and packaging and transmission of data transmitted from a network and consumer equipment layer.
- Data Processing Layer: The primary function is to handle user equipment intrusion data, including packet capture, data purification, filtering and pre-processing of databases. The job is done on fog nodes/MEC hosts for this layer.
- Detection Layer: The detection layer will be sent to the classifier to detect attacks after the intrusion data is pretreated. It uses the classification to analyze intrusion data to determine the type of attack to which it belongs. The host state of the fog nodes/MEC hosts is monitored by a safety monitoring system. At the same time, network protocols and logs are managed and documented for fog network/MEC packets. The fog node/MEC host transmits test results and the related logs to the cloud server after collection and storage of a certain volume of data. The detection layer in the MEC IDS is the main layer. In the MEC fog node/MEC host, the detection role is complete.
- Analysis Layer: The Cloud Data Center uses this layer. The key role is to evaluate the effects of the fog nodes/MEC hosts

and related logs. It will incorporate information and create such application resources into expertise. This layer, for example, will create a fog node/MEC host security status report and store it in the cloud server.

- Management Layer: Cloud service administration primarily monitors and manages the fog node/MEC hosts' security status, decides on and answers to the intrusion detection system, and the intrusion fog node/MEC hosts data and logs which can be maintained in order to enable forensic intrusion.

The detection classification of the detection layer is the most critical aspect of the intrusion data treatment scheme. It is associated with the following approach to the intervention of the cloud service to ensure system security. Our thesis therefore focuses on intrusion detectors. Deploying detectors to fog nodes/MEC hosts can fully use fog node/MEC hosts for intrusion detection calculation capabilities and storage space. Because of the restricted computational and storage capacity of fog nodes/MEC host, however, massive data cannot be processed or stored.

The simple deployment of the fog node/MEC detector on hosts cannot comply with complex network specifications. The cloud server has a wider storage capacity than fog nodes/MEC hosts so that more training sets and selection rules can be stored on the cloud servicers. It will dynamically allocate training sets to nebulae/MEC hosts such that various nebulae/MEC hosts are unique and dynamic.

The workflow of the following steps:

Step 1: The terminal accesses the fog node/MEC host and links the fog node/MEC host. The network environment of each fog node/MEC host is unique.

Step 2: The cloud-controlled Fog node/MEC cluster ensures that the complete training collection is handled by the cloud service.

Step 3: Specify the complete cloud server training and pick a sample by the rules. The premise of sample selection is that it has a sense of the fog node/MEC host network climate.

Step 4: The chosen training collection is forwarded to the fog node/MEC host via a cloud server.

Step 5: Fulfill the fog node/MEC host training phase.

Step 6: The fog node/MEC host and terminal contact can create a data stream. Detection of intrusion on fog nodes/MEC hosts is performed.

Intrusion detection and re-time response at the fog node/MEC host layer to ensure fog nodes/MEC host security and reliability. For example, the fog node/MEC host should be able to identify the type of attacker and to send an alert to deter intrusion when external intruders strike a fog node/MEC host. A fog node/MEC host will perform the calculation task locally to ensure a short latency thanks to its computation and storage capability. On fog nodes/MEC hosts, we can use lightweight and energy-efficient intrusion detection algorithms.

3.2 SVM FOR INTRUSION DETECTION

The network services for large, heterogeneous intelligent devices are provided on fog nodes/MEC hosts in FC/MEC, as representatives of clever devices delivering services via fog nodes/MEC hosts change dynamically. This means that a fog node/MEC host will at any time release a service or link to a computer. This poses a complex challenge to FC/safety. The classification device needs to be trained with the latest targeted training collection in real time for the intrusion detection method of the Fog Node/MEC host. Many of the earlier intrusion detection classification algorithms took a long time to train. These intrusion detection systems are not suited for FC/MEC paradigms due to the reduced calculation power and storage space of the fog nodes/MEC hosts. Therefore, the operation of an LDD algorithm on fog nodes/MEC hosting must be investigated.

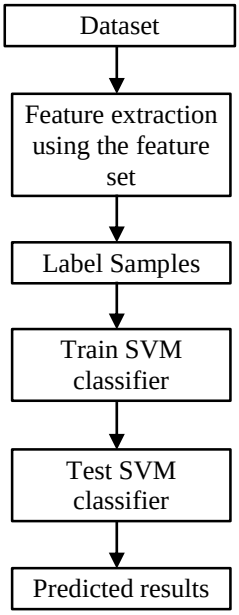

Fig.3. Proposed SVM based intrusion detection system.

The SVM was founded in the late 1970s based on the principles of statistical education. The SVM mostly addresses two-grade grading issues. A linear line or hyperplane is built as a decision limit for classification between the datasets. The closest data points to the hyperplane, which give the hyperplane construct, are called support vectors.

Algorithm: SVM Classification

Step 1: Generate signals of m-dimensions using the Poisson distribution parameter

Step 2: Extract features

Step 3: Label the normal class

Step 4: Generate signals of m-dimensions using the Poisson distribution parameter

Step 5: Extract features

Step 6: Label the intrusion class

Step 7: Concatenate the two vectors vertically

Step 8: Train SVM model using Kernel function, cross validation with samples.

4. PERFORMANCE EVALUATION

Two related words, attributes and functions, are used to prevent any confusion. The primary indicator of the arrival rate from the data to the node is called an attribute. The minimum,

maximum and median attribute derived (arrival rate for packages) are named. Thus, in this paper, two related criteria, trait and characteristics, should be considered.

A simulation is used to include the data collection to demonstrate the reliability of the proposed IDS scheme. It consists of 100 natural samples and 100 samples intruded. Reading the sensor in one unit time is a snapshot or measurement. For example, a raw sample represents a reading vector for packet arrival times. The 100 samples are the 100 vectors at 100 different times and the vector has numbers that reflect the arrival rates for the packet. The word raw here refers to the reading from the sensor without preprocessing, i.e. extraction of features. Each vector has a length, N, which is the number of elements per observation directly linked to the size of the instant period. A vector is produced by simulation to obtain a normal sample assuming that, with the Poisson distribution, the packet arrival rate reads instantly without attack/intrusion. The vector obtained with a distribution of Poisson is equally regarded as an intruding class sample, i.e., the network is being attacked. 100 raw samples are thus collected from each class. For each experiment, the simulation parameters are varied. N is represented by the number of elements per sample, and w is the length of a window that will be explained and used for the next tests of 2 to 4. The three characteristics or a combination of any two of those characteristics derived from a single vector are a pre-processed sample or observation.

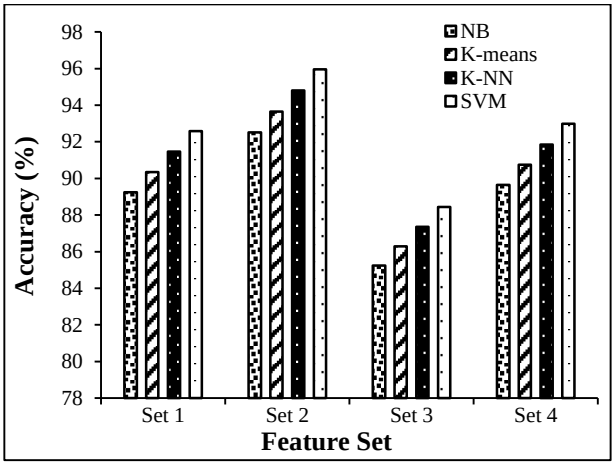

Fig.2. Accuracy

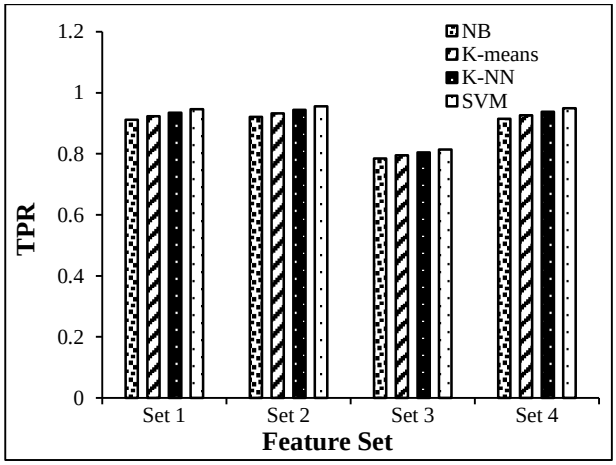

Fig.3. True Positive Rate

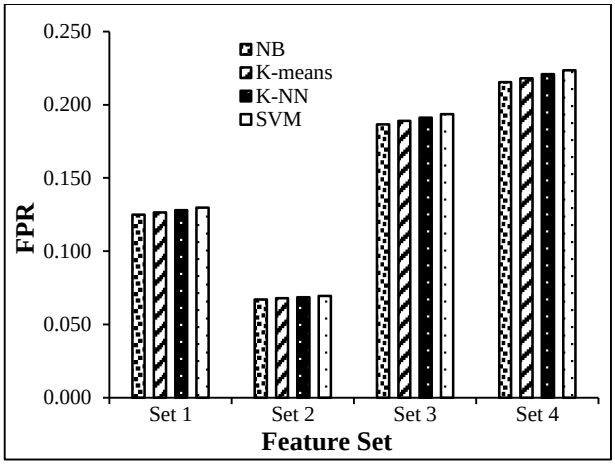

Fig.4. False Positive Rate

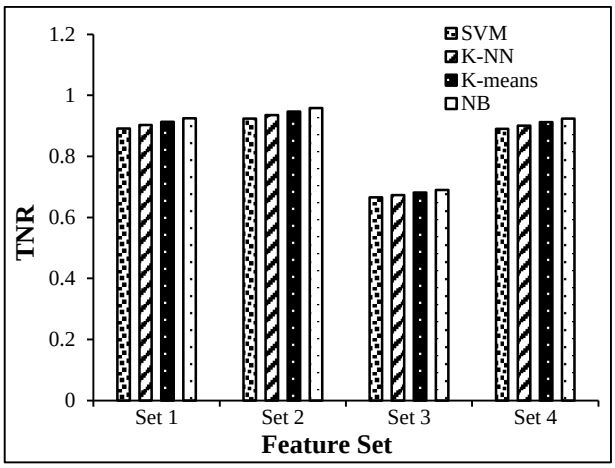

Fig.5. True Negative Rate

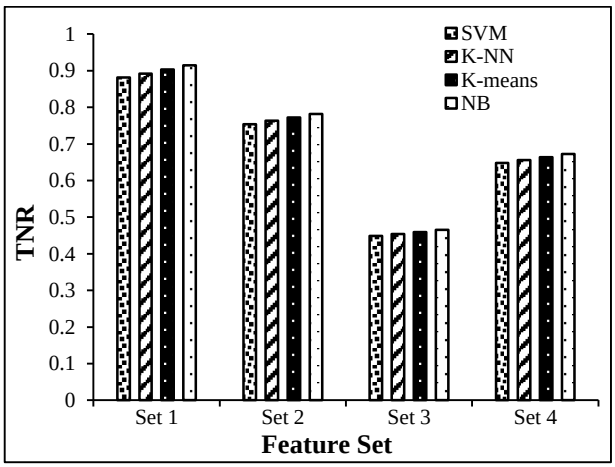

Fig.6. False Negative Rate

A SVM-based classifier with three core functions (linear, polynomial, and radial) is trained for initial simulations. Four feature variations are medium and maximum (max), medium and medium, max, medium, and medium, max and medium. The parameter of the distribution of Poisson is 2.2, while a parameter of 2.4 is used to obtain the intruded signal. The duration and the number of elements per observation of a single observation is 500. The dataset contained 100 observations per class and 40

observations per class, while the others were tested using the 60 observations.

The results of the SVM polynomial kernel feedback metrics. The polynomial kernel-based SVM performing performance evaluation outperformed all other function combinations to achieve 92% precision. The precision in all functional configurations, relative to linear SVM, was nevertheless degraded. TPR, FPR, TNR and FNR, with a median discrepancy of 0.099 and linear SVM, have almost the meme values of one single function set, with the exception of peak and median. TPR was reduced from 0.866 to 0.3, but from 0.7333 to 0.1833. The FPR in this instance was reduced of this mixture of characteristics, TNR and FNR are identical to the linear SVM.

5. CONCLUSION

In this article, we considered the SVM intrusion detection classification in MEC. However, chosen features such as medium, maximum and median, and the suggested detection scheme will be used regardless of the delivery type, provided that the traffic intensity to the node is affected by an interference or attack. The SVM is preferred for its efficient efficiency among other machine-learning algorithms. The performance analysis provided in the experimental findings supports our argument between different machine learning algorithms. In addition, the experimental findings demonstrate the lightweightness of the proposed algorithm. In addition, a comparative study of SVM's classification with other machine-based classifiers has been introduced to demonstrate the benefit of using SVM with regard to the exactness compared to other techniques.

REFERENCES

[1] Mehmet Eren, Todd P. Boren, Nitin K. Singh, Burook Misganaw, David G. Mutch, Kathleen N. Moore and Floor J. Backes, "Sparse Feature Selection for Classification and Prediction of Metastasis in Endometrial Cancer", *BMC Genomics*, Vol. 18, No. 3, pp. 233-243, 2017.

[2] C. Weinreb, S. Wolock and A.M. Klein, "Spring: A Kinetic Interface for Visualizing High Dimensional Single-Cell Expression Data", *Bioinformatics*, Vol. 34, No. 7, pp. 1246-1248, 2018.

[3] W. Ayadi, M. Elloumi and J.K. Hao, "A Biclustering Algorithm based on a Bicluster Enumeration Tree: Application to DNA Microarray Data", *Biodata Mining*, Vol. 2, No. 1, pp.1-9, 2009.

[4] M. Gill and D. Singh, "ACO Based Container Placement for CaaS in Fog Computing", *Procedia Computer Science*, Vol. 167, pp. 760-768, 2020.

[5] S.A.A. Naqvi, N. Javaid and H. Butt, "Metaheuristic Optimization Technique for Load Balancing in Cloud-Fog Environment Integrated with Smart Grid", *Proceedings of International Conference on Network-Based Information Systems*, pp. 700-711, 2018.

[6] M.K. Hussein and M.H. Mousa, "Efficient Task Offloading for IoT-Based Applications in Fog Computing using Ant Colony Optimization", *IEEE Access*, Vol. 8, pp. 37191-37201, 2020.

[7] S. Zahoor, S. Javaid and N. Javaid, "Cloud-Fog-Based Smart Grid Model for Efficient Resource Management", *Sustainability*, Vol. 10, No. 6, pp. 2079-2092, 2018.

[8] J.A. Hartigan, "Direct Clustering of a Data Matrix", *Journal of the American Statistical Association*, Vol. 67, No. 3, pp. 123-129, 1972.

[9] A. Sancetta, "Greedy Algorithms for Prediction", *Bernoulli*, Vol. 22, No. 2, pp. 1227-1277, 2016.

[10] R. Aakash and M. Nishanth, "Data Mining Approach to Predict Forest Fire using Fog Computing", *Proceedings of International Conference on Intelligent Computing and Control Systems*, pp. 1582-1587, 2018.