International Journal of Information Retrieval Research
Volume 12 • Issue 3

Detection Mechanism Using Transductive Learning and Support Vectors for Software-Defined Networks

Gaganjot Kaur, Manav Rachna University, India*
Prinima Gupta, Manav Rachna University, India
Yogesh Kumar, Indus Institute of Technology and Engineering, Ahmedabad, India

ABSTRACT

SDN has come up as a promising technology for a future network as a logically centralized controlled framework along with its physically distributed architecture isolating the control plane from sending data moving the entire choice capacity to the regulator. SDNs are turning out to be significant because of scalability, adaptability, and testing. As SDN needs overhead for operation, it makes it a target of distributed denial of service (DDoS) attacks. The extensive review in the existing literature survey provides results for small footprint of dataset causing over fitting of the classifier. In the survey, it has also been observed that the KNN-based algorithms to detect DDOS attacks are lazy learners resulting in the noisy data. This paper proposes a dual probability transductive confidence machines and support vector machine (DPTCM-SVM) classifier to avoid the over-fitting for detecting DDoS in SDN. The results generated for detection are more than 98% for all the attack classes making it an eager learning system which requires less learning space unlike the lazy learning systems.

KEYWORDS

Distributed Denial of Service (DDoS), Dual Probability Transductive Confidence Machines (DPTCM), Software-Defined Networking (SDN), Support Vector Machine (SVM), Transductive Confidence Machines (TCM)

INTRODUCTION

Now a days, SDN (Shin, M. K., Nam, K. H., & Kim, H. J, 2012) (Kim, H., & Feamster, N. 2013) is gaining huge attraction from both the field of industry and academics area by emerging as a new network management concept. The traditional distributed network management paradigm has been replaced by the centralized control platform. SDN put forwards many prospects and methods for new network management. One essential objective of SDN is to permit a system controller, called the control plane, to ignore and deal with the whole system by designing steering instruments for hidden switches. The switches, additionally called the information plane, are exclusively liable for information sending as indicated by their sending tables. Directing calculation and system the board, are dealt with by the controller. The principal motivation behind the software defined network is to move data starting with one plane then onto the next. SDN switches are constrained by a network operating

DOI: 10.4018/IJIRR.300293 *Corresponding Author

system (NOS) framework that gathers data utilizing the API and controls their sending plane, giving a theoretical model of the organization geography to the SDN regulator facilitating the applications.

Open Network Foundation (Devlic, A., John, W., & Sköldström, P, 2012) presents an elevated level engineering for SDN which are primarily part into three fundamental useful layers including framework layer, control layer and application layer as appeared in Fig. 1. It contains three principle layers: the foundation layer which bolsters the information plane activity, the control layer, and the application layers which are:

1. **Infrastructure Layer (Southbound):** Also referred as data plane, it comprises basically of Forwarding Elements (FEs) including physical switches, for example, virtual switches and Open vSwich. These switches are accessible by methods for an open interface to switch and forward groups (Devlic, A., John, W., & Sköldström, P, 2012).
2. **Control Layer:** The Control Layer is otherwise called the control plane, it comprises of a lot of programming based SDN controllers giving a merged control usefulness through open APIs to manage the system sending conduct through an open interface. Three correspondence interfaces permit the controllers to connect: southbound, northbound and east/westward interfaces (Devlic, A., John, W., & Sköldström, P, 2012).
3. **Application Layer (Northbound):** It primarily comprises of the end-client business applications that expend the SDN interchanges and system administrations. Instances of such business applications incorporate system virtualization, versatility the board, and security application, etc. (Devlic, A., John, W., & Sköldström, P, 2012).

The figure 2 furthermore includes two recognized reference application programming interfaces, called Northbound API and Southbound API (Cui, L., Yu, F. R., & Yan, Q, 2016) (Hoang, D. B., & Pham, M, 2015). The Northbound API is the interface introduced to the SDN applications' architects. It intends to open simple to-use huge level reflections prepared to cover the multifaceted nature trademark in the central framework topology and framework wide states, similarly as to release the framework official from the need to oversee low level framework center points' course of action nuances (Hoang, D. B., & Pham, M, 2015). Right now for identification of DDoS attacks we are generally worried about the Southbound API, which is the interface that utilizes state-full SDN

Figure 1. Architecture of SDN (Devlic, A., John, W., & Sköldström, P, 2012)

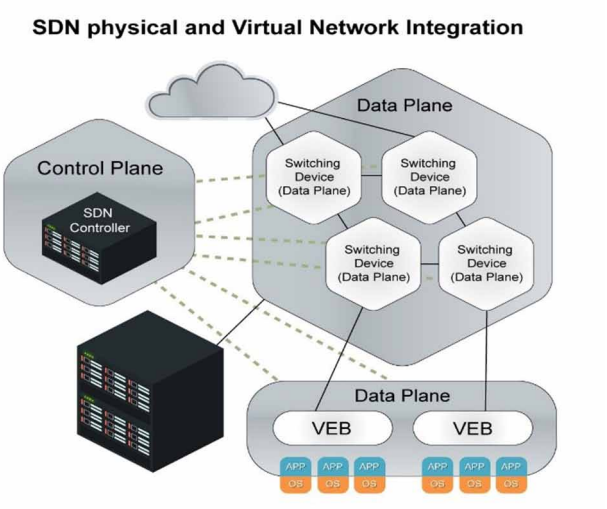
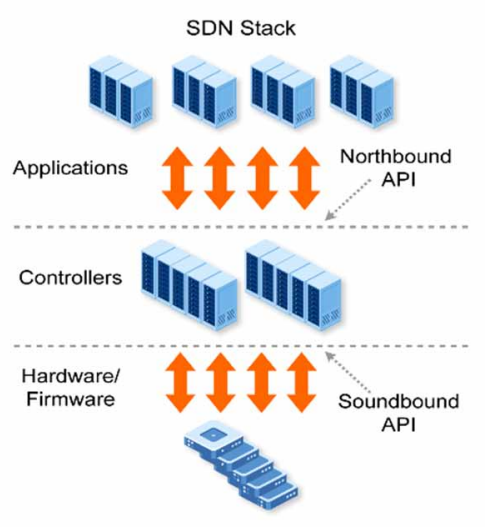


Figure 2. SDN functional layers (Cui, L., Yu, F. R., & Yan, Q, 2016)



information plane. For the most part, SDN depends on the way that the easiest capacity of a switch is sending parcels dependent on a lot of rules. One goal of SDN is to perform organize undertakings, which can't be finished without extra programming, by giving open client controlled administration of the sending equipment of a system component (Klöti, R., Kotronis, V., & Smith, P, 2013). Another goal is to convey some portion of the system intricacy from the equipment based system gadgets to the product based controller. To condense, the significant value of SDN is sensibly developing a brought together controller through the partition of control plane and information plane, which empowers the unified control and rearranges organize the executives. OpenFlow show (Klöti, R., Kotronis, V., & Smith, P, 2013) is sole part of the SDN building that mostly allows changes to permeate streams level control. It was proposed to systematize the correspondence between the switches and the item based controller in SDN designing and to enable examiners to run preliminary shows in sort out. SDNs empower us to helpfully program the system and to consider the making of dynamic stream strategies. The blend of the worldwide or framework wide view and the framework programmability supports a method of social affair knowledge from existing Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), for example, trailed by examination and concentrated reevaluating of the framework. This philosophy can render the SDN more grounded to noxious attack than regular frameworks. Even though SDN brings out huge benefits by separating both of the control and data plane from each other (Jain, R., & Paul, S, 2013), the SDN and DDoS attacks relationship is highly contradictory (Yan, Q., Yu, F. R., Gong, Q., & Li, J, 2015) (Wang, B., Zheng, Y., Lou, W., & Hou, Y. T., 2015). Denial of Service is an attempt made to entirely interrupt or corrupt availability of service/resources to authorized/genuine user by attackers (Wang, B., Zheng, Y., Lou, W., & Hou, Y. T, 2015). DDoS attacks are anything but laborious to dispatch, yet hard to prepare for so as to dispatch a compelling. DDoS attack, digital attackers frequently build up a system of PCs, which is known as a botnet. DDoS attacks can be grouped into two classes dependent on the focused on convention level: Network/transport-level and Application level. The Network or Transport level DDoS flooding attacks are attacks that have been generally propelled utilizing TCP, UDP, ICMP and DNS convention parcels and spotlight on upsetting genuine client's availability by depleting casualty system's transfer speed. Be that as it may, Application level DDoS flooding attacks center on upsetting authentic clients' administrations by depleting the server assets (e.g., Sockets, CPU, memory, plate/database transmission capacity, and I/O transfer speed). Customary DDoS attacks barrier instruments face

numerous difficulties in distributed computing conditions. Some famous examples of DOS attacks are ARP attacks, cache poisoning, tear drop, SYN flood, SHORK, black holes, SMURF, ping of death, land, finger, BOM, octopus etc. (Ankali, S. B., & Ashoka, D. V, 2011). The distributed version of DOS attacks is DDOS attack where attackers hire hundreds/thousands of compromised hosts called zombies against one target (Yan, Q., & Yu, F. R, 2015). To secure the network from DDOS attack, there are two research challenges that SDNs need to address:

- How to identify all types of DDoS attacks, example -ping of death, SYN flood, BOM, tear drop, octopus, SMURF, etc. effectively?
- How to remove the unwanted traffic without disrupting legitimate traffic?

It must be noted that the facilities provided by SDN helps to identify and respond to these attacks and these good capabilities of SDN may benefit in defeating DDoS attacks (Hoang, D. B., & Pham, M, 2015). Firstly, via decoupling control and data plane and hence making it capable to smoothly create enormous scope assault and guard tests. Secondly, by using its incorporated controller and perspective on the system which progressively isolate confirm authentic hosts and other assaulting host (Hoang, D. B., & Pham, M, 2015). And thirdly permitting programmability of the system by outside applications which permits more smart algorithms which can be smoothly implemented according to multiple DDoS attacks (Hoang, D. B., & Pham, M, 2015). Therefore, there have been various attempts to solve DDoS attacks in SDN by various researchers in the past, a summary of recent advances has been carried out in next section. By understanding the customization capabilities of SDN controller, the DDoS attacks against the SDN controller can be identified and removed by exploiting machine learning algorithms provided it is well trained on large dataset of attack patterns (Cui, L., Yu, F. R., & Yan, Q, 2016) (Jain, R., & Paul, S, 2013). This paper proposes a new effective classifier DPTCM-SVM for detecting DDoS attack in SDN using combined Transductive Confidence Machines (TCM) and Support Vector Machine (SVM). Rest of the paper is divided in five sections. Section 2, provides the general idea of related works in SDN networks for anomaly detection and DDoS attacks. Section 3, describes the methodology of the proposed DPTCM-SVM models. Section 4, provides the algorithm design of DPTCM-SVM. Materials, methods and results along with comparison between the existing and proposed model in terms of Mean Squared Error, False Positive Rate, True Positive Rate, Area under Curve and Accuracy for SDN flow classification are presented in Section 5. Conclusion and future scope is presented in Section 6.

RELATED WORKS

DDoS attacks and its various detection methods in it is an extensive research matter. Also, SDN being a relatively new topic, a little research is done in the area of its security concerns. Many recent papers (Yan, Q., Yu, F. R., Gong, Q., & Li, J, 2015) (Wang, B., Zheng, Y., Lou, W., & Hou, Y. T, 2015) (Ankali, S. B., & Ashoka, D. V, 2011) pointed out that the SDN controller is prone to DDoS attacks and suggested policing packets destined to the controller.

For intrusion detection, many Machine Learning (ML) and Data Mining (DM) techniques have been suggested and are huge hits. For example, in (Seufert, S., & O'Brien, D, 2007) focused on building ML techniques for automatic security against DDOS attacks, they worked by first distinguishing between legitimate and malicious by setting up sifting physically by hand. As separating physically is regularly outlandish errand because of the huge number of hosts engaged with the assault, they at that point created Machine Learning systems in for programmed barrier against DDoS attacks utilizing Artificial Neural Network systems (ANN) in the guard against DDoS attacks. In (Suresh, M., & Anitha, R, 2011) evaluated various ML algorithms for detecting DDoS attacks. They utilized the chi-square and Information gain highlight choice instruments for

choosing the significant characteristics. With the chose properties, different AI models, similar to Navies Bayes, C4.5, SVM, KNN, K-implies and Fuzzy c-implies bunching are produced for proficient location of DDoS attacks. Their trial results show that Fuzzy c-implies grouping gives better precision in recognizing the attacks. For inquire about CAIDA informational collection is utilized as the assault information and dependent on chi-square and data increase positioning, important highlights have been chosen. In (Yuan, X., Li, C., & Li, X, 2017) a profound learning based technique for recognizing DDoS assault was finished. They proposed a profound learning based DDoS assault location approach called Deep Defense which can consequently remove elevated level highlights from low-level ones and addition ground-breaking portrayal and surmising. They structured a repetitive profound neural system (RNN) to take in designs from arrangements of system traffic and follow organize assault exercises. The test results exhibit a superior presentation of our model contrasted and customary AI models. They had the option to lessen the mistake rate from 7.517% to 2.103% contrasted and traditional AI technique in the bigger informational collection. As Most of the present ML-based DDoS revelation approaches are under two orders: controlled and independent. Overseen ML approaches for DDoS recognizable proof rely upon openness of checked framework traffic datasets. While, independent ML approaches perceive assaults by analyzing the moving toward framework traffic. The two methodologies are tested by enormous measure of system traffic information, low location precision and high bogus positive rates. In (Idhammad, M., Afdel, K., & Belouch, M., 2018) an online consecutive semi-directed ML approach for DDoS recognition dependent on arrange Entropy estimation, Co-grouping, Information Gain Ratio and "ExraTrees" calculation. The solo piece of the methodology permits decreasing the immaterial typical traffic information for DDoS discovery which permits diminishing bogus positive rates and expanding precision. Though, the directed part permits lessening the bogus positive paces of the unaided part and to precisely characterize the DDoS traffic. An Age old Question "Can ML approaches are successfully used in real world networks opposing DDoS attacks?" was answered in (Bakker, J. N., Ng, B., & Seah, W. K, 2018). Also which ML techniques can be effectively used to identify DDoS attacks were elaborated in the paper? They have indicated how measurable order can be conveyed utilizing SDN to recognize DDoS attacks. Three classifiers were chosen their work in a disconnected situation. These were then assessed on a physical system proving ground utilizing DDoS assault situation. While factual characterization can be sent utilizing SDN to order traffic, cautious thought must be made to pick classifiers that bring about the littlest conceivable parcel handling overhead. Despite the fact that the classifiers didn't show a high Accuracy, results suggested that specific factual grouping strategies can arrange organize traffic under ordinary conditions. The transductive dependability estimation process created by Kolmogorov has its hypothetical establishments in the algorithmic hypothesis of haphazardness was done in (Noh, S., Lee, C., Choi, K., & Jung, G, 2003) (Proedrou, K., Nouretdinov, I., Vovk, V., & Gammerman, A, 2002). They demonstrated that the present AI calculations as a rule need quantifies that can give a sign of how "great" the expectations are. In any event, when such measures are available they have certain drawbacks, for example, right off the bat, they can't be applied to singular test models. They frequently depend on solid hidden suppositions are not valuable practically speaking. Their expectation strategy depends on the purported algorithmic hypothesis of irregularity. Their calculation follows the transductive methodology, concerning the order of each new model it utilizes the entire preparing set to derive a standard for that specific model as it were. Conversely, in the inductive methodology a general principle is gotten from the preparation set and afterward applied to each preparation model. Hence calculation is named as "Transductive Conductive machine (TCM)" for closest neighbors (KNN). Additionally as different ML calculations yield expectations, yet don't give any trust in those forecasts. First Approach for distinguishing DDOS attacks through inductive learning was done in (Ho, S. S., & Wechsler, H, 2003). They explored the traffic rate examination (TRA) as a traffic stream investigation system and, utilizing our TRA component, broke down TCP-based system streams under DDoS attacks. Further, we

identified the DDoS organize flooding attacks utilizing the state-activity rules accumulated by AI calculations, and contrasted our recognition execution with the benchmark. The blend of traffic rate examination and flooding attacks discovery component empowers Internet assets to be sheltered and stable from the continuous flooding attacks. An Adaptive parallel tree SVM (ABSVM) classifier was created in (Burai, P., Beko, L., Lenart, C., & Tomor, T, 2014) agreement with the standard of SVM. This paper explores characterization strategies (MLC, SVM) utilizing highlight extraction can segregate among species and clones of vitality trees. The trees analyzed have comparative characteristics because of confinement of location. A paired tree SVM classifier was created as per the guideline of SVM, utilizing plane distinguishableness proportion of chosen classes. The versatile twofold tree SVM (ABSVM) gave more exact outcomes than applied multiclass SVM strategy. The essential result of this investigation was a correlation of help vector machines (SVM) characterization techniques to assess species or clones of vitality plants. For directed system interruption discovery, TCM for K-Nearest Neighbors (TCM-KNN) was created in (Li, Y., & Guo, L, 2008). They utilized a proficient system irregularity location method dependent on TCM-KNN conspire and underlined on the component based advancements. They utilize include determination and highlight weight systems to improve TCM-KNN as a promising lightweight and on-line peculiarity identification procedure both in lessening its computational expense and in boosting its location execution. A progression of analyses on notable interruption discovery dataset KDD Cup 1999 exhibit the adequacy of our techniques introduced right now. Going before TCM-KNN an improved KNN-ACO idea for interruption recognizable proof by using KDDCUP'99 (Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A, 2009) dataset was additionally evolved. In (Xu, J., Wang, L., & Xu, Z, 2020) creators dig into an upgraded immersion attack, table-miss striking attacks to demonstrate that it is more cost-effective and incredible than the conventional immersion assault. To tackle this issue they propose an effective and convention autonomous barrier system for OpenFlow systems to alleviate table-miss striking attacks. It remains between the controller stage and other controller applications, and secures the system by utilizing four practical modules: preprocessor to get the parcel touchy fields that trigger the controller to give stream rules; assault indicator to tell the sign about the event of the assault; traffic channel to recognize the assaulted ports and channel traffic dependent on recurrence; rule sweeper to dispose of vindictive standards in the switch stream table. To improve the unwavering quality of SDN a half breed profound learning based oddity recognition plot for suspicious stream discovery with regards to social mixed media is proposed in (Garg, S., Kaur, K., Kumar, N., & Rodrigues, J. J, 2019). It involves an oddity discovery module which use improved Restricted Boltzmann Machine and Gradient Descent based Support Vector Machine to identify the anomalous exercises, and End-to-end information conveyance module to fulfill exacting QoS prerequisites of SDN, i.e., high data transfer capacity, and low inertness. Additionally as of late another design for recognizing inconsistency streams under SDN was proposed to be specific DPTCM-KNN (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) in order to give answers for the imperfections present in the SDN-based stream recognition strategies. To unravel the imperfections of SDN-based stream discovery strategies, this paper structures an inconsistency stream recognition strategy for SDN. The stream assortment module of the controllers gathers the stream table data and concentrates the stream highlights, and the inconsistency recognition component preprocesses the stream includes and performs arrangement identification on the system streams with the DPTCM-KNN calculation. The author in (Prajwal S, Siddhartha M, Charan S, Girish L, 2021) presents a guard mechanism to discover attack packets that are identified using Sniffer function and when an abnormal data flow is detected the security system of Firewall gets activated. Through the prediction of all hostile behaviors, the new defense algorithm prevents possible losses like system crash, malfunctioning and also decreases the risk of attack. The outcome shows that the proposed strategy can in fact keep attacks from happening, and can viably improve the safeguard ability. In this paper, the proposed strategy let attackers imagined that there exist a portion of the resources that they can attack on the Internet by through

the dummy data displayed by "counterfeit data generator", however truth be told the objective they identified are the virtual snare planned as the snare to draw the likely attackers. This is all to shield the organization to experience the ill effects of attacks. As an overview of the literature, SDN creates an attractive tight spot by becoming a tool to overpower DDoS attacks, along with becoming a victim of DDOS attacks. (Bambang Susilo, Riri Fitri Sari, 2021) Assessed the Random Forest and Convolutional Neural Network by studying the machine learning and deep learning approaches in IOT and SDN. In their study they have checked that Convolutional Neural Network gives better accuracy for IDS 2018 muticlass classification. However the major problems that still are required to be resolved include, first-how to use of SDN's features to its best to overcome DDoS and second-how to save SDN itself from being a target of DDoS attacks. The already presented anomaly identification algorithms in literature for SDN are lacking precision and are poor in real time.

METHODOLOGY

Dataset

The original KDD Cup (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) data set is an enormous data set for intrusion detection developed by DARPA. KDD cup dataset however suffers from various problems such as redundancy, inconsistency and it is not normalized thus we have used much improved NSL-KDD (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) data set which and selected about 100K records with 42 attributes and 22 Classes as shown in Table 1. We have further enhanced the dataset by normalizing it as mentioned in previous section.

Also it should be noted that KDD datasets are very skewed towards some attack classes such as normal packets constitute of more than 56% of packets, thus any classifier can get about 56% accuracy by just marking each class as normal without even doing any learning work. However some attacks have very few instances such as load_module or buffer_overflow attack have less than 10 instances this causes the needle in the haystack problem with KDD datasets as shown in Table 2.

Figure 3 depicts the difficulty of multi class classification, when we have more than 56.24% of the data being normal class and the other classes having very few instances. For instance the load module only had 2 instances in the first 100K instances of the KDD dataset, 0.002% of the dataset. This rare nature of the samples makes it very harder for any machine learning algorithm to classify the data with high accuracy.

Due to SVM high accuracy and potential to handle high dimensional input data, SVM is selected to for improving DPTCM learning algorithm to classify KDD Dataset (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) (Wang, W., Zhang, X., Gombault, S., & Knapskog, S. J, 2009). In contrast to KNN or Decision Trees, it does not have any local minima problem, as it has less of learning parameters to select, and so generate steady and reproducible results. Also it is a supervised learning method that can partially address the over fitting problem occurred in sentiment analysis domain. Radical basis function (RBF) based kernel is used for detecting DDoS attacks for SDN Networks over KDD dataset.

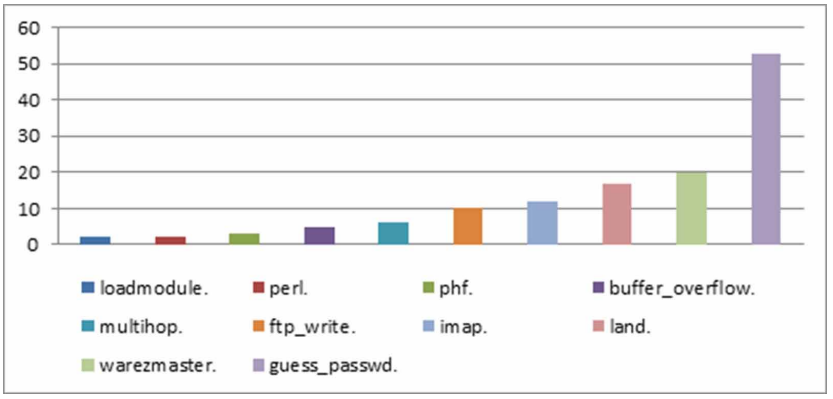
Table 1. Attack class and subclasses in Normalized NSL-KDD dataset

Attack Class	Class Names
DoS	smurf, pod, back, neptune, teardrop, land
R2L	guess_passwd, multhop, phf, imap, warezclient, warezmaster, spy, ftp_write
U2R	rootkit, loadmodule, buffer_overflow, perl
Probing	satan, portsweep, ipsweep, nmap

Table 2. Total No of instances with % of dataset in selected NSL-KDD 100K dataset

Attack Class	# of Instances	% of Dataset
loadmodule.	2	0.002%
perl.	2	0.002%
phf.	3	0.003%
buffer_overflow.	5	0.005%
multihop.	6	0.006%
ftp_write.	10	0.010%
imap.	12	0.012%
land.	17	0.017%
warezmaster.	20	0.020%
guess_passwd.	53	0.053%
teardrop.	199	0.199%
nmap.	231	0.231%
portsweep.	278	0.278%
satan.	539	0.539%
ipsweep.	760	0.760%
back.	2002	2.002%
smurf.	19143	19.143%
neptune.	20482	20.482%
normal.	56236	56.236%
Total	**100000**	**100%**

Figure 3. More than 10 classes having less than 100 instances in the dataset



For this case in the KDD dataset $U = \left\{R, \mathbb{C}_i\right\}$ where $R_i i$ is set the of collected IP packets and $\mathbb{C}_i$ is the class of the point i. Its feature points are normalized between using normalization procedure as mentioned above. The aim of SVM is to differentiate the normal and attack training classes by finding n-1 hyper plane. The Hyper plane between the 19 classes present in KDD dataset is identified

using Quadratic Programming. Using the Lagrange Multipliers theory, this QP problem is transformed and so the optimal Lagrange coefficients sets are acquired. A separating hyper plane can be described by the formula:

$$Hyperplane\left(H\right) = R \times k + \beta$$

where $k = \left\{k_1,\ \ k_2,\ldots..,\ k_n\right\}$, k_n is weight vector of n attributes and β is bias. SVM classifier formula is defined as following:

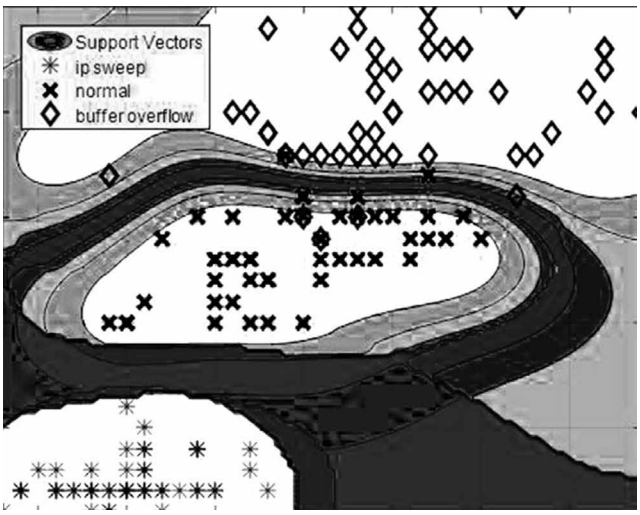
$$F\left(R\right) = \sum_{i=1}^{n} \alpha_i \gamma\left(r, R_i\right) + \beta$$

where α and β are parameters used to train the kernel $\gamma\left(r, R_i\right)$ which is given by the equation (1):

$$\gamma\left(r, R_i\right) = e^{-\left(\frac{r-r^2}{2\sigma}\right)} \tag{1}$$

although three kernels linear, polynomial, the Gaussian Radial Basic Kernel (RBF) can be used, but the RBF having the gamma γ kernel parameter and margin constant C is usually the best out of all. Separation of various class boundaries of ip sweep, buffer overflow and Normal DDoS attacks packets using RBF kernel is shown in figure 4. The blue filled area represents the boundaries of RBF kernel separating various attack classes and yellow filled area represents the various classes in consideration. It must be noted that class boundary of the RBF kernel has enough width making sure that classes fall apart easily and there is no confusion between the attacks. Also there are some packets which reside directly on top of the class boundary, these are the vector who cause the confusion of the classifier and increase errors in classification.

Figure 4. Class boundary of Support Vectors with RBF kernel



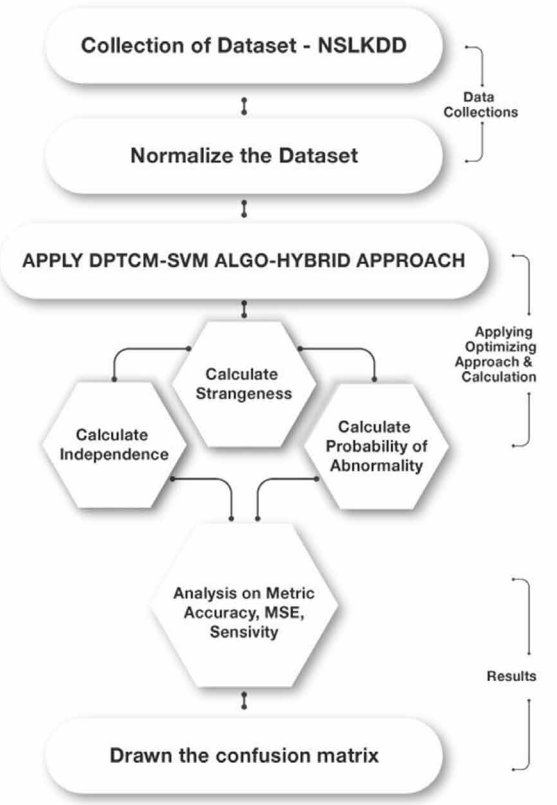
Proposed Algorithm: Dual Probability TCM-SVM Algorithm

STEP 1: Data is collected from an online repository of KDD Dataset.
STEP 2: The data is normalized and refined.
STEP 3: Proposed algorithm is implemented and parameters are calculated.
STEP 4: Analysis is done on basis of Accuracy, Mean Squared Error and Sensitivity.
STEP 5: Drawn Confusion Matrix

This paper presents design for identifying irregularity streams for SDN condition and recommends an abnormality stream discovery calculation DPTCM-KNN (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) to evacuate the impediments of the SDN-based stream recognition strategies. DPTCM-KNN utilizes the possibility of TCM (Proedrou, K., Nouretdinov, I., Vovk, V., & Gammerman, A, 2002) for example by misusing stochastic calculation to get the certainty level of a recognition point. It misuses the "likelihood P" to check if the identification point has a place with the classification. The higher the estimation of P is the higher are the odds that the location point fit into the class.

The downsides of the TCM-KNN (Noh, S., Lee, C., Choi, K., & Jung, G, 2003) algorithm are strangeness and independence in its detection, but the proposed algorithm takes "strangeness and independence" as inspection standards and improves accuracy for anomaly flow detection.

Figure 5. Flow of proposed work



However major shortcoming of KNN based algorithms is that they are lazy learners or in other words it means that it does not take anything from its own instruction specifics and rather utilizes that training data itself for classification. KNN algorithm will find the K nearest neighbors to the new instance from the training data, so as to classify the label of a new instance. The classified label will have appointed as the most common label across the K nearest neighboring points. The major drawback of KNN method is that it is time consuming. The algorithm should measure the distance at each point of prediction and must sort all the training data. This will lead to slow execution in case of large number of training examples. One more shortcoming of KNN is that it is not robust enough, as it is a lazy learner, due to which algorithm is not able to generalize well and thus making it less robust to noisy data. Moreover, altering K can alter consequential predicted class label.

Also as underlying problem is anomaly detection in SDN networks where patterns of anomalous behavior changes are rapid it makes sense to use eager learning.

That makes us consider of combining the DPTCM (Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z, 2018) (strangeness and independence as inspection standard) with eager learning algorithm especially Support Vector Machines (SVM). The major benefit achieved in implementing a keen (eager) learning method SVM, is that the objective outcome will be approximated extensively during training, which requires comparatively less space than using a lazy learning system. Keen learning systems are additionally robust, as they deal much better with noise in the training data. Keen learning is a kind of unplugged research, in which post-training queries have nil effect on the system itself, and therefore the same query will generate the same result always.

DPTCM SVM ALGORITHM

Formally, considering training set X of n elements i.e. $X = \left(x_1, C_1\right), \left(x_1, C_1\right) \ldots \left(x_n, C_n\right)$ where $X = (F_1, F_2 \ldots. F_n)$ is the set of feature values for example *j* and C_j is the classification for example j, taking values from a finite set of possible classifications, which is $\left\{1, 2, \ldots, \mathbb{C}\right\}$, where $\mathbb{C}$ is total no. of classes, in our case $\mathbb{C}$ represents the attack class for example it may be buffer_overflow, smurf, teardrop, or any normal out of 19 total attack classes. Additionally, the test set of models are used to prepare the information and accordingly the genuine orders are retained. The significant point is to assign each test model one of the potential orders. To expand insight in the predictions, the confidence measure is given for each classification and next sections explores how the confidence measure to the predictions can be assigned.

Dataset Normalization

To prepare the raw data suitable for the training, normalization process (Wang, W., Zhang, X., Gombault, S., & Knapskog, S. J, 2009) is required. Moreover, to fast the process of training neural networks normalization is highly required. Data normalization is of different types. It sometimes applied to scale the data in the same range of values for each input to reduce biasness of SVM for one feature to another. This scalability feature, fasten up the whole training process. Above all it is helpful for modeling application where the inputs are on extensively dissimilar scales. Min-max normalization of can be taken as:

$$Norm\left(X_i\right) = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

where, x_i is the current feature vector, x_{min} and x_{max} are the minimum and maximum values under feature vector x.

Distance Function

"Euclidean distance" is the widely used distance function to evaluate the distance between two points in feature space. Let P and Q are the two points represented by feature vectors $X = (F_1, F_2 \ldots F_n)$ and $B = (C_1, C_2, ..., C_n)$, where n is the dimensionality of the feature space. To compute distance between P and Q, the normalized Euclidean metric is generally used by:

$$dist\left(P,Q\right) = \sqrt{\frac{\sum_{i=1}^{n}\left(F_i - \mathbb{C}_i\right)^2}{n}} \tag{3}$$

Strangeness Measure

Strangeness measure (or nonconformity score) - p-value function is a test of confidences, for approximating an all-inclusive test for irregularity, which in its general structure is non-calculable. There are multiple definitions of strangeness measure present in literature. But, the concept behind is that it tells about the vulnerability of the fact of the matter being estimated with concerning the various marked instances of a class. The larger value of strangeness measure implies larger uncertainty. It is also described as the proportion between the entirety of the good ways from the point to its k-closest neighbors inside the class viable and the aggregate of the k-closest neighbors outside the class. The strangeness α_{iy} of a point i with regard to a class y is described by equation mentioned below:

$$\delta_{iy} = \frac{\sum_{i=1}^{k} D_{ij}^{y}}{\sum_{i=1}^{k} D_{ij}^{-y}} \tag{4}$$

Independence Measure (Φ)

The Independence measure in DPTCM-SVM classifier is the margin of error between a support point and its Φ support vectors, the L2-Norm between the two support vectors. Independence can be expressed as follows:

$$\varnothing_{\mathcal{Y}i} = \sum_{i=1}^{\Phi} Dist_{pq}^{\mathcal{Y}} \tag{5}$$

where, $\varnothing_{yi}$ - independence of input vector relative to class $\mathcal{Y}$; where Φ - support vectors and $Dist_{ij}^{\mathcal{Y}}$ is the L2 Norm or the Euclidian distance between two feature points *p* and the *q*. Similarly $\mathcal{Y}i$ represents the margin of error of the support vector point to a given class $\mathcal{Y}$. The lower the independence value makes it more likely that the support vector point is independence to the class $\mathcal{Y}$ i.e it may very likely belong to this particular class. Contrary to the measure strangeness, which is used to identify the distance between the support vector and the class, the independence is used as a measure to measure how close a given point *i* is to a class $\mathcal{Y}$ in absolute terms.

Probability of Abnormality (ρ)

The confidence measure ρ or probability of abnormality is also been used in DPTCM-SVM algorithm to identify the possibility anomaly of a support vector positional to other vector points. in other words Probability of abnormality ρ, measures the degree of belongingness to a vector point in a class. As the value of ρ increases for a class $\mathcal{Y}$ the possibility that the feature point

belongs to this specific class increases as well, the probability of abnormality for a feature vector in classifier can be calculated using:

$$\rho_1\left(\delta_i\right) = \frac{\left\{j = \left(1 \ldots n\right) : \delta_j \geq \delta_i\right\}}{n+1} \tag{6}$$

where, δ_i is the strangeness measure of feature point *i* in class $\mathcal{Y}$ and δ_j is the strangeness measure of feature point *j* in class $\mathcal{Y}$; $\left\{j = \left(1 \ldots n\right) : \delta_j \geq \delta_i\right\}$ represents for feature vector *j* the number of support vector points whose given strangeness is higher than i[th] vector in class $\mathcal{Y}$. Similarly, another probability of abnormality point ρ_2 is calculated using independence measure $\varnothing$ as:

$$\rho_2\left(\varnothing_i\right) = \frac{\left\{j = \left(1 \ldots n\right) : \varnothing_j \geq \varnothing_i\right\}}{n+1} \tag{7}$$

the ρ_1 and ρ_2 are taken as confidence measure to identify abnormality behaviors of feature point *i* and *j* in DPTCM-SVM algorithm. The feature point is considered normal if bother confidences ρ_1 and ρ_2 are normal.

Algorithm: DPTCM-SVM algorithm

Parameters: ϑ (number of Support Vectors), r Feature vectors
Input: X (KDD Dataset with η data points)
Output: Class assignment for each data point *i* in dataset ($\mathcal{Y} \in$ 19 attack classes in KDD dataset)

1. Initialize $\hat{X}$ with size of dataset X, for storing confidence updated dataset
2. **for** f=1 to r do
3. *normalize the dataset using (1) and store*
4. ***end for***
5. **for** f=1 to r do
6. calculate l2-Norm dist (f_i, f_j) distance
7. *for each class w.r.t normal class using (2)*
8. *store distances*
9. ***end for***
10. **for** i=1 to η do
11. **for** each class c = 1 to $\mathcal{Y}$ do
12. *Calculate strangeness δ_{iy} for all data points input dataset as shown in (3)*
13. Calculate Independence Measure (Φ) as shown in (iii)
14. Calculate probability of abnormality $\rho_1\left(\delta_i\right)$ and $\rho_2\left(\varnothing_i\right)$ using (5) and (6)
15. find class c such that dist of $\rho_1\left(\delta_i\right)$ and $\rho_2\left(\varnothing_i\right)$ is closest
16. store confidence values for the class with minimum distance
17. combine X with confidence values as $\hat{X}$

18. **end for**
19. train svm classifier on updated dataset $\hat{X}$ as shown in
20. and get $\hat{\mathcal{Y}} = svm\left(\hat{X}\right)$
21. evaluate using true class true class $\mathcal{Y}$ and estimated class $\hat{\mathcal{Y}}$

EXPERIMENTAL ANALYSIS

To assess the performance evaluation of DPTCM-SVM algorithm for underlying NSL-KDD dataset, several experiments have been conducted on dataset. All experiments were implemented in MATLAB R2016a(Moore, 2017)environment on a Intel pc with 8 GB RAM. For SVM kernel, Binary RBF kernel has been used which was converted into multi class classifier where attacks represented the class. Also, the values of C = 1.0 for SVM, and error tolerance of e^{-3} is used. Evaluation metrics is explained in following sections.

Evaluation

To evaluate the proposed work flowing metrics including classification Accuracy, sensitivity, specificity, confusion matrix, mean squared error (MSE) have been used. The presentation of the DPTCM-SVM classification algorithm with existing works is also analyzed.

1. **Accuracy:** Accuracy is the one of the most common method of evaluation as it can be portrayed as the proportion of the no. of effectively ordered attack and is given by the equation of the expansion of TP with TN partitioned by the complete no. of cases N:

$$Accuracy = \left(TP + TN\right) / N$$

Accuracy of the DPTCM-SVM and other algorithms with respect to sample size is displayed in figure 6. Although DPTCM-SVM is executed for more than 100K sample size, but for comparing with other algorithm we have used sample sizes as mentioned in the paper (Proedrou, K., Nouretdinov, I., Vovk, V., & Gammerman, A, 2002). It can be observed from figure 6 that DPTCM-SVM produces better accuracy as in contrast to other algorithms.

2. **Sensitivity:** Sensitivity assess the impact of uncertainty in each unsure PC contribution on a specific model yield. Sensitivity is utilized to measure the divergence between a novel perception and the current traffic. It is nonparametric choice used to suitably detect attacks. Also DPTCM-SVM possess better Classification Sensitivity which is also known as True Positive Rate (TPR) or correctly classified positive and is evaluated by the formula TP divided by the adding TP with FN:

$$Sensitivity = TP / \left(TP + FN\right)$$

It can be clearly observed from table 3 that for the different sample size the average value of the true positive rate of the proposed algorithm that is DPTCM-SVM is 4.9% better than DPTCM-KNN and 6.6% better than TCM KNN which are the lazy learning methods that means it does not learn anything from the training data and simply uses the training data itself for classification.

3. **Specificity:** Specificity which is defined as the rate of correctly classified negative to evaluate the performance of detection classifier. The experiment results in the above figure shows that the detection rate TPR is higher in the DPTSVM-KNN algorithm however it's TPR or Sensitivity

Figure 6. Accuracy of the DPTCM-SVM and other algorithms with respect to sample size

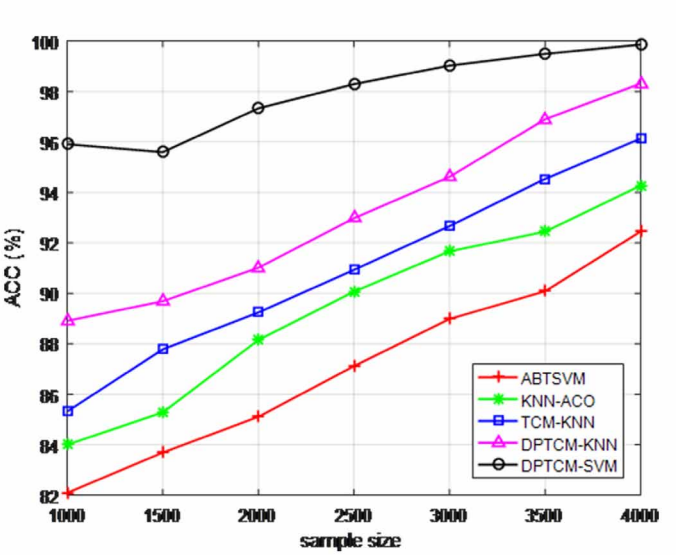
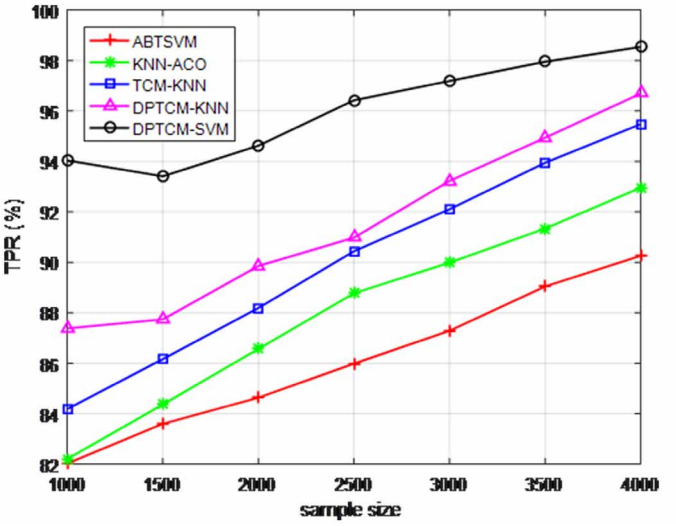


Figure 7. TPR of the DPTCM-SVM and other algorithms with respect to sample size



to a given classifier is lower that DPTSVM-SVM. TPR of DPTSVM-SVM is about 5% higher than DPTSVM-KNN. Also in case of Specificity that is given by the formula of the ratio of TN to the addition of TN with FP:

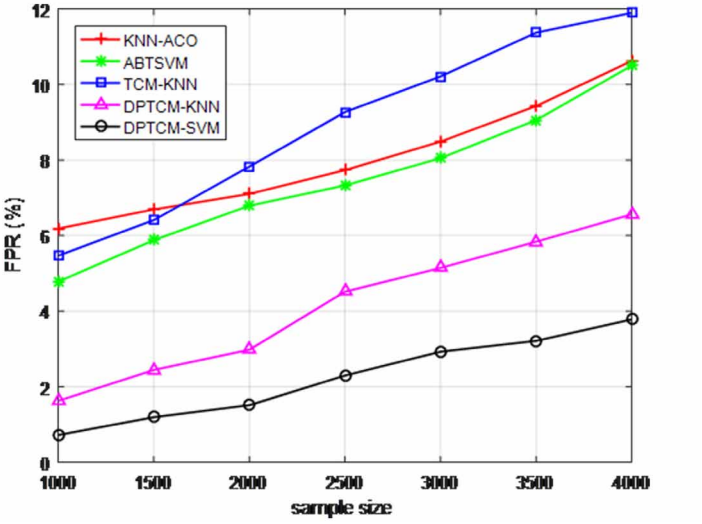
False Positive Rate = 100 - Specificity

$$Specificity = TN \,/\, \left(FP + TN\right)$$

Table 3. TPR of the DPTCM-SVM and other algorithms with respect to sample size

Samples	ABTSVM	KNN-ACO	TCM-KNN	DPTCM-KNN	DPTCM-SVM
1000	82.025	82.205	84.185	87.38	94.04
1500	83.6	84.365	86.165	87.74	93.41
2000	84.635	86.57	88.19	89.855	94.625
2500	85.985	88.775	90.44	90.98	96.425
3000	87.29	89.99	92.105	93.23	97.19
3500	89.045	91.34	93.95	94.94	97.955
4000	90.26	92.96	95.48	96.695	98.54
Average	**86.12**	**88.02929**	**90.07357**	**91.54571**	**96.02643**

Figure 8. False Positive Rate or Specificity of algorithms with sample size



DPTSVM-SVM concept performed better than all existing algorithms under consideration. Maximum FPR of DPTSVM-SVM was around <4% of instances and for DPTSVM-KNN was >6% of instances. As the dataset gets larger and larger this difference will also become great as well.

From table 4 it is observed that DPTCM-SVM is better than other algorithms in terms of errors detected as well. DPTCM-SVM has 46% less errors than DPTCM-KNN and 75% less errors than TCM-KNN. DPTCM-SVM deals much better with noise in the training data proving that it has very less flaws and so making it more accurate to use than all other algorithms.

4. **Confusion Matrix:** The confusion matrix displays the accuracy of the classifiers for the specified category in the flow of detection and implementation. In existing algorithms all instances were translated to either normal or attack class to classify as binary classification, however DPTCM-SVM algorithm is a multiclass classifier, and the confusion matrix showing TPR and FPR rates of each class is shown in figure 9. It is cleared from the figure that DPTCM-SVM produces almost near perfect >98% TPR rate for most of the attack cases under NSL-KDD 100K dataset.

Figure 9. Confusion matrix of DPTCM-SVM Classifier

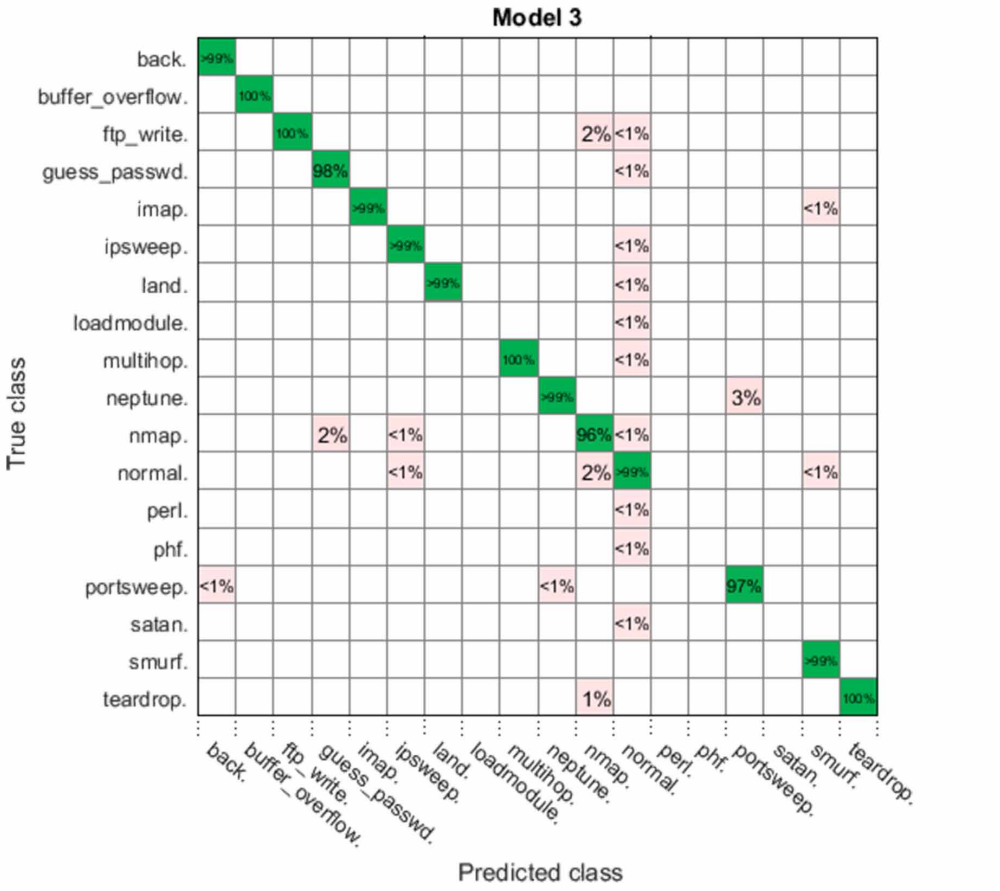


Table 4. FPR of the DPTCM-SVM and other algorithms with respect to sample size

Samples	ABTSVM	KNN-ACO	TCM-KNN	DPTCM-KNN	DPTCM-SVM
1000	6.1877	4.7814	5.4689	1.6253	0.71882
1500	6.6928	5.8804	6.4116	2.4429	1.1929
2000	7.1045	6.7918	7.8284	2.9795	1.5109
2500	7.7347	7.3283	9.2711	4.5159	2.2975
3000	8.4899	8.0524	10.214	5.1462	2.9277
3500	9.4326	9.0574	11.375	5.839	3.2144
4000	10.625	10.5	11.9	6.5627	3.7814
Total	**8.04**	**7.48**	**8.92**	**4.16**	**2.23**

The confusion matrix shown in figure 9 provides more clarity by showing how effective the DPTCM-SVM was for each class of attack, at max the classifier showed 3% miss classification for the port scan attack and the 2% for the nmap and the guess_passwd attack. For all other attacks the miss classification was 1% or less.

5. **Mean Squared Error:** Mean Squared Error of the classifier computes the average squared difference between the predicted and real value. The lesser the mean squared error value is more accurate the algorithm is. In contrast to other algorithms, the DPTCM-SVM algorithm essentially reduces the characterization error by reducing anomalous attack classification or FPR. Overall MSE was also very low (0.8%) compared to 1.8% in DPTCM-KNN. Thus proposed DPTCM-SVM provides lower FPR rates, which results in higher AUC and Accuracy rates. From Table 5 above it is evident that the DPTCM-SVM algorithm has the largest Area under the Curve (AUC) and so making more accurate as compared to other algorithms.

Results and Discussion

After performing the evaluation it's been analyzed that the proposed algorithm Dual Probability Transductive Confidence Machines (DPTCM-SVM) support vector machine has given better results for all the parameters chosen for the analysis. All the techniques that were used to detect DDOS attacks have the probability for finding errors in the approximate range of 92%~98%. Compared to the other techniques that used same dataset of NSL-KDD, DPTCM-SVM helps finding errors at appropriate rate i.e at 99% and also specifies the name of the attack among all the other classes. For all evaluation metrics the proposed algorithm provides lower false positive rate and produces mean squared error to very less minimal value at 0.8%. The transductive learning in DPTCM has improved the performance by creating a feature digest for all the features present in the dataset. When there are large numbers of features in the dataset, and the dataset itself is large this Meta information helps in the reducing time complexity of the classifier because these features (Strangeness, Independence and Abnormality) help the classifier (SVM) by assigning weightage to the instance. This transduced information can be used by any classifier in real time application to improve the performance.

Table 5. AUC, MSE and ACC of algorithms

Authors*	Applied techniques	Parameters		
		Accuracy(ACC)	Area Under Curve(AUC)	Mean Squared Error(MSE)
(Peng, 2018)	TCM-KNN	~92%	0.9207	~8%
	KNN-ACO	~93%	0.9311	~7%
	ABTSVM	~94%	0.9487	~6%
	DPTCM-KNN	98.2%	0.9788	1.8%
(Shone, 2017)	EHOMLP-IDS	87.9%	0.88	2.1%
	HADMLP-IDS	91.7%	0.94	1.1%
	MFOMLP-IDS	86.3%	0.87	1.5%
(Sumathi, 2020)	DNN	97.1%	0.97	2.8%
	DNN-CMA	98.9%	0.92	1%
Proposed System	DPTCM-SVM	99.2%	0.988	0.8%

CONCLUSION

In this research our investigation uncovered that current SDN-based answers for DDoS recognition and alleviation neglected to meet application explicit necessities for DDoS attack identification. The proposed algorithm uses the concept of Eager Learning method where in the target function is approximated globally during the training process of dataset making use of less space and giving higher accuracy with lesser error values than the lazy learning system. The Machine learning calculations used in the proposed algorithm identifies the DDoS attack with the example created from the dataset. Although SDN provides flexibility the security in SDN still remains a major concern. SDN is especially weak against the DDoS attacks as flexibility always comes with computational costs. In view of the test results performed on KDD'99 Dataset the proposed DPTCM-SVM provides much lower False Positives and higher accuracy as compared to existing schemes. Through simulations it's been proved that the superiority of the proposed DPTCM-SVM algorithm in the form of FPR, TPR, MSE and accuracy which comes due to combination of DPTCM with a strong classifier in contrast to a weaker one such as KNN previously used in literature. DPTCM-SVM compared with the other algorithms, significantly reduces the classification error by reducing anomalous attack classification. In terms of classification accuracy the proposed algorithm achieves highest accuracy (99.2%) than the other algorithms that we have analyzed. Not only that the proposed algorithm is able to reduce false positive rate more than 75% and is 5% better in true positive rate. The limitation of the study is that the proposed algorithm is run on predefined dataset. The proposed work is implemented on single NSL KDD dataset. Evaluation of the proposed algorithm on different dataset is not carried out. It would be really worthwhile to investigate performance of DPTCM algorithm under real time scenario.

FUTURE SCOPE

For our future works, the work will be done on refining of the DPTCM-SVM method for identifying the real time DDoS attacks by collecting and filtering immediate network packets, for this will focus on how the proposed solution can be directly incorporated via POX Controller in MININET (Kaur, S., Singh, J., & Ghumman, N. S, 2014). The work can also be done in development of DPTCM-SVM based intrusion detection system to add more Security and functionality to the proposed system. Also instead of SVM newer deep learning approaches can also be explored for more accuracy of the system. The proposed algorithm in real time application can be used in firewall for security and also it can be used as an algorithm to implement DOS safe routing through routers.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Ankali, S. B., & Ashoka, D. V. (2011). Detection architecture of application layer DDoS attack for internet. *International Journal of Advanced Networking and Applications*, *3*(1), 984.

Ashraf, J., & Latif, S. (2014, November). Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques. In *2014 National Software Engineering Conference* (pp. 55-60). IEEE.

Bakker, J. N., Ng, B., & Seah, W. K. (2018, July). Can machine learning techniques be effectively used in real networks against DDoS attacks? In *2018 27th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-6). IEEE.

Burai, P., Beko, L., Lenart, C., & Tomor, T. (2014, June). Classification of energy tree species using support vector machines. In *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)* (pp. 1-4). IEEE.

Cui, L., Yu, F. R., & Yan, Q. (2016). When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network*, *30*(1), 58–65.

Devlic, A., John, W., & Sköldström, P. (2012, October). A use-case based analysis of network management functions in the ONF SDN model. In *2012 European Workshop on Software Defined Networking* (pp. 85-90). IEEE.

Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446–452.

Garg, S., Kaur, K., Kumar, N., & Rodrigues, J. J. (2019). Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective. *IEEE Transactions on Multimedia*, *21*(3), 566–578.

Ho, S. S., & Wechsler, H. (2003, July). Transductive confidence machine for active learning. In *Proceedings of the International Joint Conference on Neural Networks*, 2003 (Vol. 2, pp. 1435-1440). IEEE.

Hoang, D. B., & Pham, M. (2015, September). On software-defined networking and the design of SDN controllers. In *2015 6th International Conference on the Network of the Future (NOF)* (pp. 1-3). IEEE.

Idhammad, M., Afdel, K., & Belouch, M. (2018). Semi-supervised machine learning approach for DDoS detection. *Applied Intelligence*, *48*(10), 3193–3208.

Jain, R., & Paul, S. (2013). Network virtualization and software defined networking for cloud computing: A survey. *IEEE Communications Magazine*, *51*(11), 24–31.

Kaur, S., Singh, J., & Ghumman, N. S. (2014, August). Network programmability using POX controller. In *ICCCS International Conference on Communication, Computing & Systems* (Vol. 138). IEEE.

Kayacik, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2005, October). Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets. In *Proceedings of the third annual conference on privacy, security and trust* (*Vol. 94*, pp. 1723-1722). Academic Press.

Kim, H., & Feamster, N. (2013). Improving network management with software defined networking. *IEEE Communications Magazine*, *51*(2), 114–119. doi:10.1109/MCOM.2013.6461195

Klöti, R., Kotronis, V., & Smith, P. (2013, October). OpenFlow: A security analysis. In *2013 21st IEEE International Conference on Network Protocols (ICNP)* (pp. 1-6). IEEE.

Li, Y., & Guo, L. (2008, March). TCM-KNN scheme for network anomaly detection using feature-based optimizations. In *Proceedings of the 2008 ACM symposium on applied computing* (pp. 2103-2109). ACM.

Noh, S., Lee, C., Choi, K., & Jung, G. (2003, March). Detecting distributed denial of service (ddos) attacks through inductive learning. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 286-295). Springer.

Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018). A detection method for anomaly flow in software defined network. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 27809–27817.

Prajwal, S., Siddhartha, M., Charan, S., & Girish, L. (2021). *DDOS Detection and Mitigation SDN using Support Vector Machine*. Academic Press.

Proedrou, K., Nouretdinov, I., Vovk, V., & Gammerman, A. (2002, August). Transductive confidence machines for pattern recognition. In *European Conference on Machine Learning* (pp. 381-390). Springer.

Seufert, S., & O'Brien, D. (2007, June). Machine learning for automatic defence against distributed denial of service attacks. In *2007 IEEE International Conference on Communications* (pp. 1217-1222). IEEE.

Shin, M. K., Nam, K. H., & Kim, H. J. (2012, October). Software-defined networking (SDN): A reference architecture and open APIs. In *2012 International Conference on ICT Convergence (ICTC)* (pp. 360-361). IEEE.

Shone, Ngoc, Phai, & Shi. (2017). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Sumathi, S., & Karthikeyan, N. (2020). Detection of distributed denial of service using deep learning neural network. *Journal of Ambient Intelligence and Humanized Computing*.

Suresh, M., & Anitha, R. (2011, July). Evaluating machine learning algorithms for detecting DDoS attacks. In *International Conference on Network Security and Applications* (pp. 441-452). Springer.

Susilo & Sari. (2021). *Intrusion Detection in Software Defined Network using Deep Learning Approach*. Academic Press.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1-6). IEEE.

Wang, B., Zheng, Y., Lou, W., & Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and software-defined networking. *Computer Networks*, *81*, 308–319.

Wang, W., Zhang, X., Gombault, S., & Knapskog, S. J. (2009, December). Attribute normalization in network intrusion detection. In *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks* (pp. 448-453). IEEE.

Xu, J., Wang, L., & Xu, Z. (2020). An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Computer Networks*, *169*, 107092.

Yan, Q., & Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Communications Magazine*, *53*(4), 52–59.

Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys and Tutorials*, *18*(1), 602–622.

Yuan, X., Li, C., & Li, X. (2017, May). Deep Defense: identifying DDoS attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8). IEEE.

Gaganjot Kaur has been appointed as an Assistant Professor in Department of Computer Science and Technology. Her educational qualification includes Regular M.Tech from Punjab Technical University Jalandhar and B.Tech from Institute of Engineering & Technology Bhaddal both in CSE. She is currently pursuing Ph.D in Computer Science from Manav Rachna University. She has entire 10 years of experience in academics. She is the author of 22 published papers and 1 book chapter in Scopus and 7 are published in International Conferences and 2 are published in National Conferences and 11 are the part of International Journals and 2 are published in National Seminars. She is responsible for successfully carrying out both research and teaching duties. She also has MS Azure Fundamentals and Azure Data Fundamentals certificate to her credit and has 08 certificate for successfully completing NPTEL courses.

Prinima Gupta is presently working as Professor in CST Department, at Manav Rachna University, Faridabad. She has over 15.6 years of experience including academics & research. She was awarded PhD in the area of "Ad-hoc Networks" from Singhania University, Rajasthan in 2013. She completed her MCA from N.C Engineering College, Israna, Panipat in 2006 and M.Tech from Maharishi Dayanand University, Rohtak in 2015. Her Research Area includes Ad-hoc networks, Information Security, Data Mining & Graphics. She has published more than 25 papers in National & International level journals and 2 Book Chapters. She has also presented 15 papers in National & International Conferences and also attended various workshops & FDPs. She is currently guiding 06 Ph.D. research scholars and guided 02 M.Tech Dissertations in 2015. She worked with various journals and Conference as reviewer of the paper and a member of technical committee. At MRU, Dr. Prinima Gupta is handling the responsibility of Ph.D Program coordinator and also working as overall Coordinator of Research Program in DoCST.

Yogesh Kumar is working as Associate Professor at Indus Institute of Technology & Engineering, Rancharda, Ahmedabad. He has done his Ph.D. CSE from Punjabi University, Patiala. Prior to this, he has done his M.Tech CSE from Punjabi university, Patiala. He is having total 14 years of experience including teaching and research. His research areas include Natural Language Processing, Computer Vision, Machine Learning and Data Science.