Modeling of Hybrid Intrusion Detection System in Internet of Things using Support Vector Machine and Decision Tree

John Kwesi Amfo
Kwame Nkrumah University of Science and Technology
Department of Computer Science

James Ben Hayfron-Acquah
Kwame Nkrumah University of Science and Technology
Department of Computer Science

ABSTRACT

The Internet of Things (IoT) system, according to literature is prone to many attacks from its interconnected devices. Most of these security treats that IoT is likely to encounter have been identified. These attacks and other security issues have called for the modeling and implementation of algorithms that can identify the current and emerging intrusion and vulnerabilities in IoT so that best security preventive methods may be deployed against them. In this study, a new algorithm, is proposed; cascading Decision Tree (DT) algorithm and Support Vector Machine (SVM) algorithm to improve classification of attacks and consequently the security systems. The proposed algorithm used Support Vector Machine for selection of features based on correlation in the features of the Network Socket Layer - Knowledge Discovery Data (NSL-KDD) data sets. And classification of intrusions was based on the DT algorithm, due to its performance over the SVM. The result of the proposed algorithm proved that the classification of attacks in the decision tree algorithm is improved in terms of prediction speed and training time. In addition, it enhanced the performance of Decision Tree in classifying misclassified classes such as “rootkit” in intrusion detection.

Keywords

Internet of Things, Intrusion Detection, Support Vector Machine, Decision Tree, Feature Selection.

1. INTRODUCTION

Internet of Things (IoT) is a sensible network connecting all things to the web to exchange info with in agreement of protocols [10]. In the IoT, the things could be someone with a farm animal that has a biochip transponder, an implant for heart monitoring, automobile with built-in sensors that alerts drivers when there is low pressure in tire - or other man-made or natural objects that is possible to assign an IP address which is given the capability to transfer packets over networks. IoT came up from the convergence of micro services, micro-electromechanical systems (MEMS), wireless technologies as well as the Internet.

In the IoT network, objects are connected wirelessly with good sensors. Therefore, information can be accessed by anyone, from anywhere and at any time [3]. In the IoT network, objects are connected wirelessly with good sensors. These devices will move with one another while there is no human intervention [3]. In IoT, distinctive addressing schemes are used to move with alternative objects or things and work with these objects to make new services or applications. Varied applications such as good homes, good cities, health observation, good setting, and good water [1], are introduced in IoT.

Among several alternatives or importance of IoT applications, the security of IoT systems cannot be left unconcerned. The development of the “Internet of Things” is changing the reality and the fast increase in security treats in related products. It requires high security to enhance good communication and transaction control between embedded devices on the network [9]. IoT devices can be accessed anywhere through untrusted network just like the web, therefore IoT networks according to literature are prone to many attacks from its interconnected devices most of which are outlined in Ren [8]. In the study of Le et al.,[2], two novel types of RPL attacks were presented: local repair attack and rank attack. In addition, the spread of Internet applications and its high usage quality result in vital increasing of cyber-attacks. Some of these Cyber-attacks on IoT applications include sinkhole attack, wormhole attack, sybil attack, denial of service attack, etc [5],[7] which are categorized into four main classes namely: denial of service (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), and Probing Attack.

According to Pongle and Chavan [6], security provision against such attacks in IoT is a challenge since these devices have fixed resources, lossy communication links, and uses a collection of novel technologies in IoT like 6LoWPAN and RPL. As a result, it becomes less difficult to attack in IoT network. In addition, detecting attacks against IoT has as well become a hard problem as it was in the beginning of computer age (mid-1990s) to solve in the network security field.

Network security has become a matter of importance and several other ways are developed for these attacks. Consequently, attack detection and other security issues in IoT have called for the modeling and implementation of algorithms that can identify the current and emerging intrusion and vulnerabilities in IoT so that best security preventive methods may be deployed against them. For this purpose, Intrusion detection system (IDS) is being employed to observe the attacks that may occur on IoT device networks. Data mining Techniques, Machine Learning, Neural networks, Collective Intelligence, evolutionary algorithms and statistical ways are a number of algorithms that are used for classification, coaching and reviewing detection accuracy with analysis primarily based on the standard datasets in Intrusion Detection Systems.

In this research, the hybrid algorithmic program is introduced based on decision tree and Support Vector Machine (SVM) exploitation feature selection and decision rules to use on IDS. The main plan is to use the strengths of each algorithms to boost detection, enhance the accuracy and scale back the speed of error detection of the results. During this algorithmic program, the simplest options are selected by SVM, after decision tree is employed to form selections and outline rules.

The results of applying projected algorithmic program are analyzed on the quality dataset KDD Cup99. The projected technique guarantees high detection rate that is proved by simulation results.

2. PRELIMINARIES OF THE SVM AND DT ALGORITHMS

2.1 The Support Vector Machine Method

In today's usage of machine learning, support vector machine is thought in concert of the strongest and most correct ways within the machine learning algorithms. SVM is among the supervised learning ways that is used for classification, prediction and regression. This technique is a comparatively new approach that in recent years has shown sensible performance for classification compared to older ways like perceptron neural networks and it is easy as well. This formula was introduced in 1998 by Vapnik [11]. SVM because of its sensible ability to generalize and being superior to different algorithms in classification and regression is extremely widespread. SVM in theory is intended for binary classification; therefore, its method forward to solve the existing classification issues between traditional and abnormal or suspicious behavior is appropriate in follow pattern audit [11]. SVM base on decision planes concept to define the boundaries. Hyperplanes are constructed in a high-dimensional space with an SVM, separating all the data points of one category from the other. The hyperplane with the largest margin between the two classes for an SVM is the best. It mostly happens that classes cannot be separated linearly. Due to this, the original finite dimensional space is mapped into a much higher-dimensional space, which makes separation easier by using what is referred as "kernel trick".

Though we can apply SVM to variety of optimization problems like regression, the most challenge is that of classification of data. SVM identifies the data points as being negative or positive, and the problem is to find an optimal hyper-plane separating the data points with higher margin.

In solving classification problem with the concept of SVM, mathematically let assume a vector $\bar{\boldsymbol{w}}$, perpendicular to the media line of the hyper-plane (street) and $\bar{\boldsymbol{u}}$ an unknown quantity vector in the plane. The vector $\bar{\boldsymbol{u}}$ is projected to $\bar{\boldsymbol{w}}$ to identify whether it is on the positive or negative side of the hyperplanes. This gives

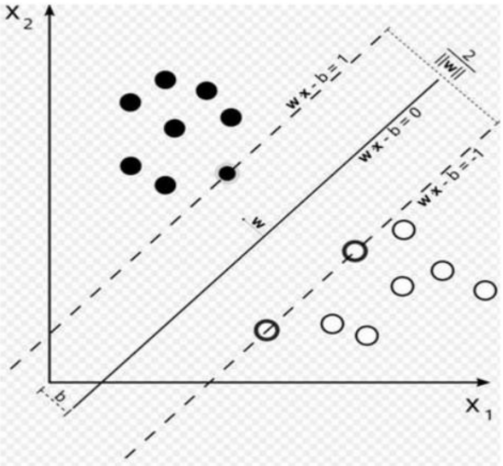

Fig. 1. The Street and boundaries of SVM

$$\bar{u} \cdot \bar{w} \geq c \qquad (1)$$

where c is a scalar which determines whether $\bar{u}$ is on the negative or positive side of the street.

In addition, to identify whether $\bar{u}$ is a positive or negative sample, let choose a positive x_+ and negative sample $x_-$and let $\mathcal{Y}_i$ = ± such that

$$\begin{aligned} yi(\bar{w} \cdot \bar{x}\ +) &\geq 1\, if\ \bar{u}\ is\ a + ve\ sample \\ yi(\bar{w}.\bar{x}\ \_) &\geq 1\, if\ \bar{u}\ is\ a - ve\ sample \end{aligned} \qquad (2)$$

Hence, let expect for negative and positive samples

$$y_i(\,(\bar{w}.\bar{x}\ + b) - 1\ \geq 0 \qquad (3)$$

which implies

$$yi(\,\bar{w}.\bar{x}i\ +\ b) -\ 1\ =\ 0 \qquad (4)$$

for samples on the boundaries.

Again, let x_+ and x_- be samples on the boundaries of both the positive and negative samples respectively. Hence, let expect that

$$(x_{+\,-\,x_-})\frac{\bar{w}}{|\bar{w}|} = \frac{2}{|\bar{w}|} \qquad (5)$$

Where $\frac{\bar{w}}{|\bar{w}|}$ is a unit vector normal to the plane and $(x_+\ - x_-)$ is the distance (width) between the two boundaries. Since $\bar{w}x_+ = 1 - b\ and\ \bar{w}\,x_-$ from equation (4). Hence the objective of the problem is

$$Max\quad \frac{1}{2}\,||\bar{w}||^2 \qquad (6)$$

Subject to

$$yi(\bar{w}.\bar{x}i\ +\ b) -\ 1\ =\ 0 \qquad (7)$$

To solve this problem, the standard method is to use Lagrange theory to convert the problem to a dual Lagrange problem. Using Lagrange, it obtain

$$L = \frac{1}{2}\ ||\bar{w}||^2 - \textstyle\sum \alpha_i\ [\,y_i(\bar{w} \cdot \bar{x}_{\mathrm{I}} + \mathrm{b}) - 1] \qquad (8)$$

where α_i are Lagrange multipliers. Finding $\frac{\partial \mathrm{L}}{\partial \bar{\varpi}}$ and $\frac{\partial L}{\partial b}$ and setting to zero yields respectively

$$\bar{w} = \sum \alpha_i\ \ y_i x_i \qquad (9)$$

and

$$\sum \alpha_i\ \ y_i = 0 \qquad (10)$$

Hence for optimum solution, let obtain,

$$L = \sum \alpha_i\ - \frac{1}{2}\sum\sum \alpha_i\ \alpha_i\ \ y_i y_i x_i \cdot x_i \qquad (11)$$

The dual problem is the following

$$Min\ \ \phi(\,\bar{a}\,) = \sum \alpha_i\ -\ \alpha_i\ \alpha_i\ \ y_i y_i x_i \cdot x_i \qquad (12)$$

with conditions

$$\sum \alpha_i\, y_i = 0 \qquad (13)$$

and

$$a_i\ \geq 0 \qquad (14)$$

2.3 The Decision Tree

One classification algorithm in knowledge mining that has brought a vital step forward in the field of artificial intelligence, learning systems, non-parametric statistics and knowledge mining is the Decision tree (DT) algorithm. The DT algorithm uses divide and conquer technique in constructing decision trees of data set features recursively. Two methods i.e. bottom up or top down manner is used to build Decision tree. With empty root node, Decision trees are constructed and then nodes which corresponds to the algorithms are built [12],[13]. The partitioning of the training data set is done and then a tree-like structure is created. An information gain (IG) for each attribute is computed for selection purposes. Attributes with highest value are considered best for decision trees.

Let D be a training dataset containing *m* unique classes, $C_i(i = 1,2...m)$. C_i,D set of tuples in class C_i in *D*. $|C_i,D|$ and $|D|$ represent respectively the number of tuples in C_i,D and *D*, and Pi is the probability that a tuple in *D* is also in C_i, and is evaluated by $|C_i,D|/|D|$. Feature *A* is selected to partition *D* into *v* groups D_j $i = 1,2,...,v$ where D_j contains tuples in *D* with a_j of *A* as an outcome. A method is employed to select the best or splitting features as follows:

$$info\ (D) = -\sum_{\mathrm{i}-1}^{\mathrm{m}} \mathrm{P}\log\ \ \mathrm{Pi} \qquad (15)$$

$$Gain\ (a,T) = Entropy(a) - Info\ (a,T) \qquad (16)$$

$$Info(a,T) + \frac{Ta,v}{Ta}\ Entropy\ (av) \qquad (17)$$

$$GainRation\ (a,T) = \frac{Gain(a,T)}{Split\ Info(a,T)} \qquad (18)$$

$$Split\ Info(a,T) + = -\ \frac{|T_a,v|}{|T_a|}\log\frac{T_a,v}{T_a} \qquad (19)$$

The complex tree is a typical decision tree algorithm which is an advanced form of the decision tree developed by Ross Quinlan [12]. C4.5 supervised learning classifier or is also known as statistical. The decision tree formula has several deserves and demerits. Decision tree reveals all the relations between rules, so it makes understanding the information structure simple, it describes generated rules and all existing categories in the coaching information. It has easy calculations however a vast memory is needed to store the whole tree and mining the foundations [15].

3. DATASET FOR THE STUDY

Providing input data is the first step of data mining method in every Intrusion Detection System (IDS). Such data are provided by different methods from various resources. To detect the anomalous behavior in network (not in host), the best resource we have is the network traffic which is the sending packets between the origin and the destination. In KDD Cup 99 dataset, the network traffic of the host or the network is collected. [14], in his study using the NSL-KDD, which is the advanced version of KDD Cup 99 to detect intrusion, observed that the number of records in the NSL-KDD test and train datasets are reasonable. This advantage, he observed, makes it less costive to do the analysis on the whole dataset without randomly sampling from it. In addition, Manhod et al [4], in his study conducted a statistical analysis on the KDD Cup 99 data set, and found two important issues highly affecting how the evaluated systems perform, and also result in a very poor evaluation of anomaly detection processes. The first issue outlined in the KDD dataset is the higher number of redundant records. Secondly, the statistics of both KDD test and train datasets shows that random parts of the KDD train set are used as test sets. Hence, a new dataset, NSL-KDD, was proposed consisting of selected records of the complete KDD dataset which suffers not from any of the shortcomings outlined. As a result, the NSL-KDD dataset for intrusion detection was used in this experiment. The data set contains 22 attack types which are put into four main classes namely:

i. Denial of service (DOS): With this attack type, attackers make some memory resources or computing too full or too busy in handling legitimate requests, or denies users access to a machine. Examples are Land, Back, neptune, pod.

ii. Remote to user (R2L): With this attack type, an attacker without an account on remote machines sends a packets to the machine over networks and takes advantage of some vulnerability to obtain local access to the machine. Examples include Ftp write, Dictionary, Guess passwd, Imap.

iii. User to root (U2R): In this case, attackers start out with access to a normal account user on the system and is capable to take advantage of the vulnerabilities of the system to gain access as root user to the systems. Examples include Fdformat, Loadmodule.

iv. Probing: With this attack type, an attacker scans the computers on the network to find known vulnerabilities or gather information. Attackers that are available on a network with a map of machines and services can use the information gathered to take advantage of the system. Examples include Nmap, Ipsweep, Satan.

Each record in this dataset has 42 features. Generally, features have three forms including continuous, discrete and symbolic which have different range of values.

Table 1. NSL-KDD Dataset (Small Set)

Total number of records in original dataset	**Training data set**	**Testing data set**
2022	1011	1011

4. THE CONFUSION MATRIX

The confusion matrix is a table of values used in describing how well a classification model (or" classifier") performs on data set with known true values. It comes with terminologies used in analysing the performance of the model. To fully understand the terms in connection with the confusion matrix, we consider for example the confusion matrix for a binary classifier.

Table 2. The Confusion Matrix

No	50	10
Yes	5	100
	No	Yes

True Class

Predicted Class

Assuming that we are to classify patients as whether they have a particular disease or not. Then from the matrix above, its learn that

i. It have 2 actual and predicted classes each.

ii. The classifier made 165 predictions, 110 times 'YES' and 55 times 'NO'.

iii. However, in actual sense, 105 patients have the disease and 60 no disease.

The Confusion matrix explains how the various algorithms and their training set performed during the training. The green portion of the matrix indicate the percentage of the various attacks that were detected correctly without errors. The red portion in the matrix identifies the features that were confused during the training.

Terminologies with the Confusion Matrix

i. **Accuracy:** This is how often the classifier is correct. It is the capability of a classifier to measure correctly the intrusions from the training data set. This is the ratio of data that is classified correctly to the total data classified. It is given by

$(TP + TN)/TOTAL. Total = TP + FP + TN + FN$

ii. **Misclassification Rate:** This is how often the classifier predicts wrongly. It is also called the error rate. It is given by $(FP + FN)/TOTAL$ which is equivalent to $1-Accuracy$.

iii. **True positive rate (TPR):** This shows the performance of the classifier per class. It is the ratio of *TP* to the actual YES. Thus *TP/*actual YES

iv. **False Positive Rate (FPR):** The effectiveness of the variety of existing models is obtained using FPR. It is a major concern in setting up a network A normal packet is classified as attack or abnormal data type. It is defined as; $FPR = FP/$ (*Actual* YES). *Actual* YES = $TN + FP$

v. **False negative rate (FNR):** This shows the performance of the classifier per class. This is given by $1 - TPR$.

5. THE ROC CURVE

The ROC curve is a plot of true positive rate against false positive rate for the trained classifier that is selected. The red marker on the plot indicates the classifier's performance.

The position of the marker illustrates the FPR values and that of TPR for the classifier selected.

For example, in Figure (5), the marker shows a false positive rate (FPR) of 0.02 indicating the selected classifier assigns incorrectly 2% of the data to the positive class. Also, a true positive rate of 0.83 shows that 83% of the data was assigned correctly by the selected classifier to the positive class.

A red marker positioned at right angle to the top left of the plot indicates perfect results. In contrast, poor results no better than random is a line at 45 degrees. The Area Under Curve (AOC) number measures the classifier's overall quality. Larger AOC values show better performance by the classifier.

6. IMPLEMENTATION AND ANALYSIS OF RESULTS

The objective of this work is to implement a hybrid DT-SVM algorithm to improve classification of attacks and consequently the security systems in IoTs. The proposed algorithm used Support Vector Machine for selection of features based on correlation in the features of the Network Socket Layer - Knowledge Discovery Data (NSL-KDD) data sets. And classification of intrusions was based on the DT algorithm, due to its performance over the SVM. The performance of Decision Tree algorithm over the Support Vector Machine was justified through a comparative study of the two algorithms using the same NSL-KDD data sets. The comparison of the algorithms was done with the measures from the Confusion Matrix and the ROC curve. To access the performance of the two algorithms for classification, they were both run using the same data sets ((NSL-KDD)) and the results shown in figures (2 to 4) and table (3).

From the confusion matrix shown in 2 and 3), it can be observed in figure 2 that, "smurf" and "teardrop" were 100% classified by DT and also 99% of "neptune" was correctly classified. This indicates that DT algorithm is most effective in classifying the intrusion classes; "smurf", "teardrop" and "neptune" as shown in model 1 (CDT) of figure 2. This is because it has highest probability of detecting these attacks without or with very less errors.

Comparatively, the dataset was run with the SVM algorithm and the output shown in the confusion matrix, figure 3. From the output, it is observed that 98% of "neptune" was correctly classified, which appeared to be the only class detected with little error. Also, 76% and 82% of "smurf" and "teardrop" was correctly classified as compared to the DT which was 100%.

From the values of the true positive rates (TPR) for both DT and SVM algorithm, it can be observed that DT algorithm has higher probability in classifying intrusion classes with very little error compared to SVM algorithm.

Above all, the general performance of the two algorithms, DT and SVM were examined in terms of Accuracy, Prediction speed and Training time. This is illustrated in table 3. From the table (table 3), it is observed that, the DT algorithm is more accurate with higher prediction speed and smaller training time compared to the SVM algorithm. This implies that, the DT algorithm performs better to the SVM.

In addition, ROC curve (figure 4) indicates that model 1 (DT algorithm) assigns 100% of the observations correctly to the true.

Table 3. Performance of DT and SVM in terms of Accuracy, Prediction speed and Training time.

Algorithm	Accuracy (%)	Prediction speed (obs/sec)	Training Time (sec)
Complex DT	96.2	7400	8.1844
SVM	94.3	510	8.493

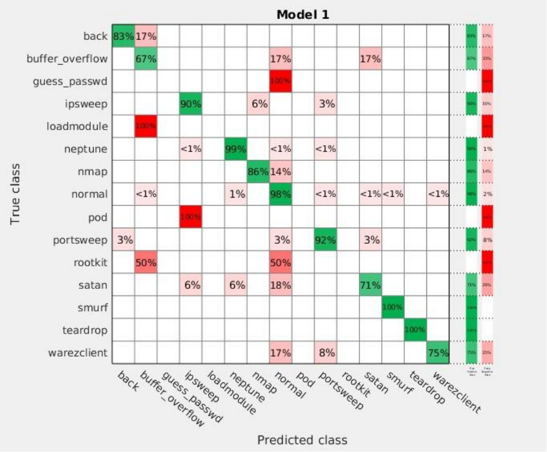

Fig. 2.True positive rates for DT on NSL-KDD Data set

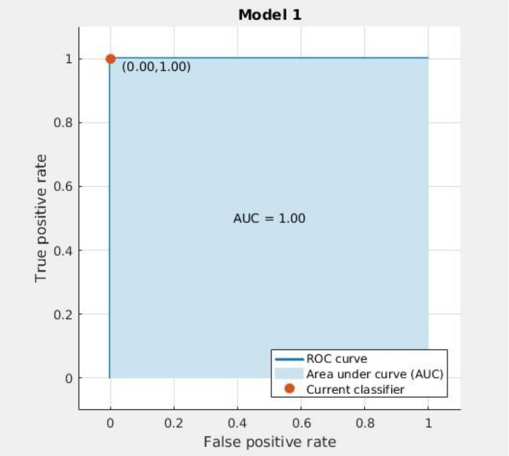

Fig. 4.ROC Curve showing the performance of DT

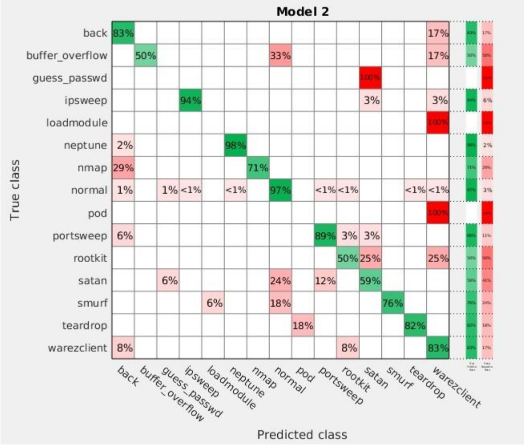

Fig. 3.True positive rates for SVM on NSL-KDD Data set

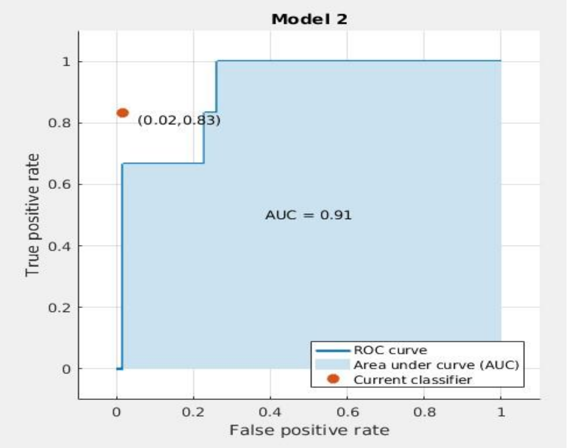

Fig. 5.ROC Curve showing the performance of SVM

class compared to SVM which assigns 83% of the observations correctly to the true class. In addition, SVM classifies the intrusion class with 2% false positive rate while the DT recorded 0% false positive rate. An indication of DT performing better in intrusion detection than the SVM.

Due to the performance of the DT algorithm, the proposed model used the DT algorithm after feature selection with SVM. The feature selection was based on correlation in the features of the NSL-KDD data sets. The highly correlated feature variables were removed and the end result trained and classified using the DT algorithm.

The overall performance of the cascaded SVM-DT model for classification was studied in terms of Accuracy, Prediction speed and Training time. This is illustrated in table 4. In addition, the ROC curve was plotted.

From table 4, it is observed that, the lesser the features for classification, the less training time required. With accuracy, it was shown that, it decreases marginally with decreasing number of selected features.

The algorithm was run on a different machine (AMD A10-5750M APU with Radeon(tm) HD Graphics 2.50Ghz, 6.00 GB Ram, 64bit Windows OS) and the result compared to the main results which was run on “5.8GB, Intel Pentium(R) CPU B940 @ 2.00gHZ, Linux 64-bit OS, Intel Sandy bridge Mobile Graphics”. The following results was observed.

Table 4. Performance of cascaded SVM-DT model in terms of Accuracy, Prediction speed and Training time (Linux OS).

Selected Features	Accuracy (%)	Prediction speed (obs/sec)	Training Time (sec)
21 Features	94.7	7200	1.6554
30 Features	95.1	8800	1.6787
36 Features	95.5	6800	1.6970

Table 5. Performance of cascaded SVM-DT model in terms of Accuracy, Prediction speed and Training time. (Windows OS)

Selected Features	Accuracy (%)	Prediction speed (obs/sec)	Training Time (sec)
21 Features	94.2	6100	1.6215
30 Features	95.3	5500	1.7048
36 Features	96.2	5100	1.7405

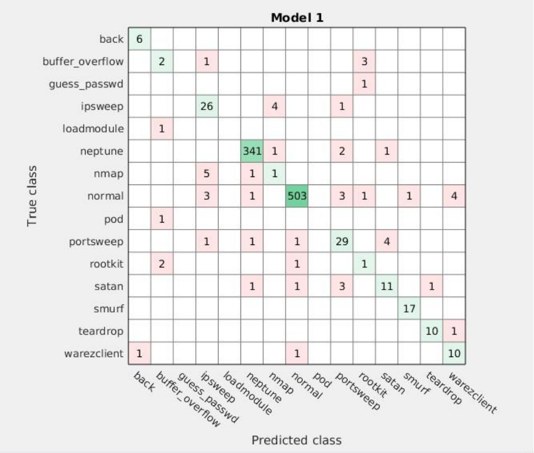

Fig. 6. Performance of cascaded SVM-DT model with 21 selected features.

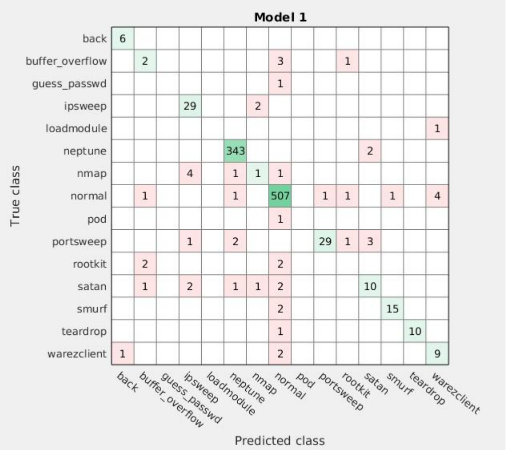

Fig. 7. Performance of cascaded SVM-DT model with 30 selected features

The result obtained from the output of the two machines shows consistent results. However, it is observed that the accuracy, prediction time and the training time is machine dependent.

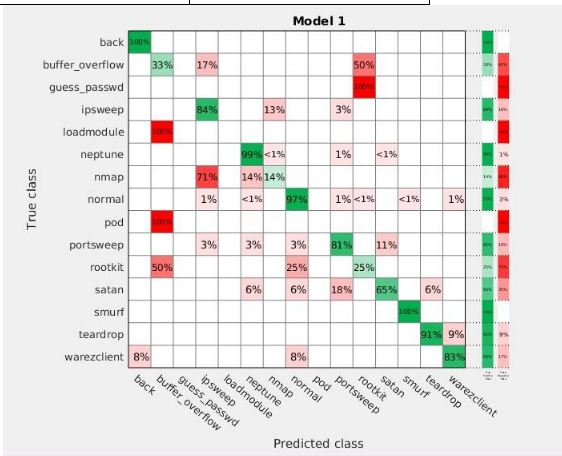

Fig. 8.True positive rates of cascaded SVM-DT model with 21 selected features.

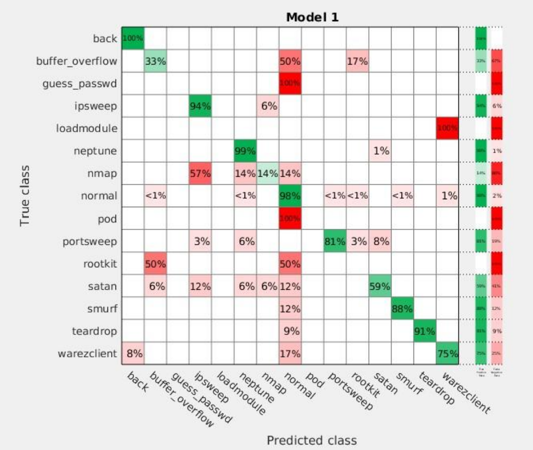

Fig. 9.True positive rates of cascaded SVM-DT model with 30 selected features

The ROC curve for the cascaded SVM-DT model with selected features is as shown in figure 10. It is observed that, the ROC curves for selected 21 and 30 features are similar. Both curves show 100% true positive rate (correctly assigned features) for the DT algorithm.

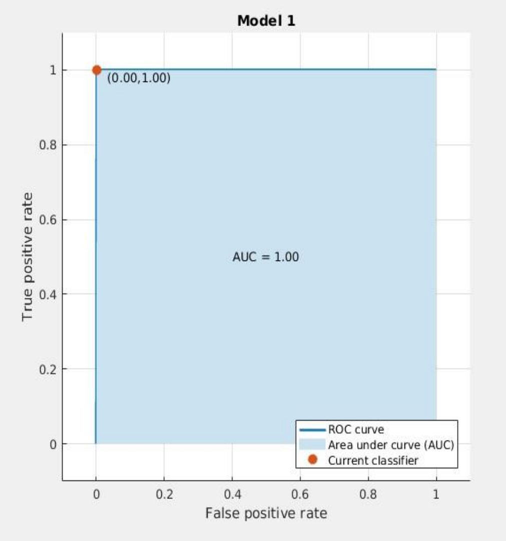

Fig. 10. The ROC Curve of the cascaded SVM-DT model with 21 selected features

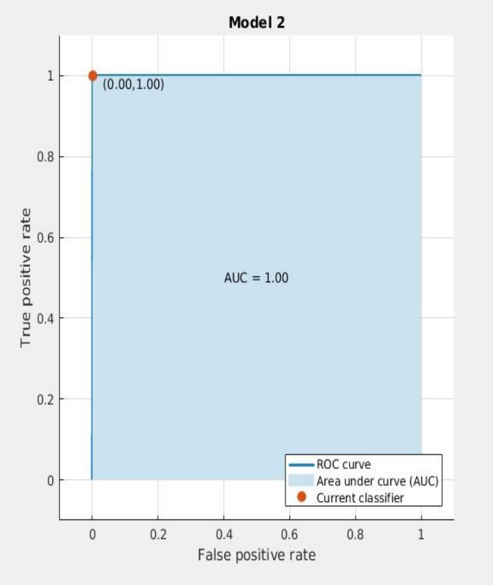

Fig. 11. The ROC Curve of the cascaded SVM-DT model with 30 selected features

7. CONCLUSION

Intrusion detection in Internet of Things (IoT) is of much interest to researchers. Hence, recent research focus on improving existing algorithms in order to improve the security of IoT systems by detecting intrusion attacks and reducing false alarms. In this study, a new algorithm, was proposed, cascading DT algorithm and SVM algorithm to improve classification of attacks and consequently the security systems.

In order to achieve this, the performance of both DT algorithm and SVM algorithm were extensively studied. From literature, the NSL-KDD data set as compared to the original KDD data set was found to address some key issues with the original KDD data set. To analyze the performance of these algorithms, the NSL-KDD data set was used for training and testing. From the results of the two algorithms, it was observed that the Decision tree algorithm outperformed the Support Vector Machine. This was made evident in terms of their accuracy, training time and the prediction speed (table 3).

In this study, a new algorithm was proposed for classification. From the model proposed, classification of intrusions was based on the DT algorithm, due to its performance. The Support Vector Machine was used for the selection features which was based on correlation in the features of the NSL-KDD data sets. The highly correlated feature variables were removed and the end result trained and classified using the DT algorithm. For different values of standard deviation, three different feature selections were done based on the SVM and the result classified using the DT algorithm. From the results obtained, it was observed that features such as “rootkit” which was 100% misclassified in the original SVM and DT algorithm was 25% detected with the proposed model which included feature selection. Which informs us, certain intrusion classes which are mostly misclassified can be detected by correctly selecting the right features for classification. In addition, a significant improvement in the proposed algorithm in comparing with the DT algorithm was observed. The training time was reduced from 8.1844sec to 1.6554sec. The prediction speed for some selected number of features was observed to have increased and it decreased for some selected number of features. From this results, it can be observed that, computational complexities in terms of time possibly prediction speed has been reduced.

The result of the proposed algorithm proved that the classification of attacks in the decision tree algorithm is improved in terms of prediction speed and training time. In addition, it enhanced the performance of DT in classifying misclassified classes such as “rootkit” in intrusion detection. Hence intrusion detection with feature selection for classification of attacks in IoT systems should be considered.

8. RECOMMENDATION

From the findings of this study, it’s recommended that intrusion detection with feature selection for classification of attacks in IoT systems should be considered. Since this in effect improves attack classification in terms of prediction speed and training time. Finally, irrespective of the improvement achieved in the performance of the existing DT algorithm by including the feature selection, the proposed algorithm still has some weaknesses such as the classification accuracy. The accuracy of the proposed algorithm, even-though decreased in a small percentage, further studies can be carried out to improve the proposed algorithm in terms of accuracy.

9. REFERENCES

[1] Gokul P., Sreeram S., Chandra M. and Reddy S. (2015). Development of Industrial Intrusion Detection and Monitoring Using Internet of Things. *International Journal of Technical Research and Applications,* 84-89.

[2] Le A., Loo J, Luo Y, and Lasebae A., (2011) Specificationbased IDS for securing RPL from topology attacks. *IEEE*, 978-1-4577-2028-4/11.

[3] . Li S., Li Da Xu, and Zhao S. (2015). The Internet of things: a survey. *Springer Information Systems Frontiers,* 17(2), 243259

[4] Mahbod T., Ebrahim B., Wei L., and Ali A. G. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. *National Research Council, Canada*

[5] Okan C., and Ozgur K. S. (2015). A Survey of Intrusion Detection Systems in Wireless Sensor Networks. *6th*

International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), 40-43

[6] Pongle P., and Chavan G. (July 2015.). Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *International Journal of Computer Applications*, 1-9.

[7] Rahaman A. S., Rashmi R. S., Moutushi S., Souvik S., Jamuna K. S., and Koushik M. (2014). Intelligent Intrusion Detection System in Wireless Sensor Network. *Advances in Intelligent Systems and Computing* 328. Springer, 707-711.

[8] Ren, L. (2015). IoT security: problems, Challenges and Solution. *Santa Clara, CA*, 1-32

[9] Schneier, B. (2014). Schneier on security. Retrieved August 18, 2016, from https://www.schneier.com/blog/archives/2014/01/ security risks 9.html

[10] Shanzhi C., Hui X, Dake L., Bo H. and Hucheng W. (2014). A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective. *IEEE Internet of Things Journal*, 1(4), 349 - 359.

[11] Shojaie, Z. (2005). Algorithms and Advanced Data mining Concept. Jahad Daneshgahi, Amirkabir University of Technology. Tehran, 4-9.

[12] Snehal S. S., and Daivshala R. D. (2016). Improved Intrusion Detection System using cascading of C4.5 Decision Tree and Support Vector Machine. *International Journal of Emerging Technology and Advanced Engineering* , 6(8), 167-169.

[13] Vaishali K. and Sangita S. C. (2014). Improved Intrusion Detection System using C4.5 Decision Tree and Support Vector Machine. *International Journal of Computer Science and Information Technologies,* 5(2), 1463-1467

[14] Vaishali Kosamkar (2013) Improved Intrusion Detection System using C4.5 Decision Tree and Support Vector Machine, *Master's Thesis, Department of Computer Engineering, University of Mumbai*, Retrieved from: www.academia.edu/16645429/Vaishali mam report

[15] Wu, S. Y. (2009). Data mining-based intrusion detectors. *Expert Systems with Applications,* 56(1), 36