



An Efficient Method to Decide the Malicious Traffic: A Voting-Based Efficient Method

Ajay Kumar, Bharati Vidyapeeth (Deemed), Institute of Management and Research, New Delhi, India

 <https://orcid.org/0000-0003-0919-1395>

Jitendra Singh, PGDAV College, University of Delhi, India*

Vikas Kumar, Central University of Haryana, India

 <https://orcid.org/0000-0002-6753-1557>

Saurabh Shrivastava, Bundelkhand University, Jhansi, India

ABSTRACT

To address the high rate of false alarms, this article proposed a voting-based method to efficiently predict intrusions in real time. To carry out this study, an intrusion detection dataset from UNSW was downloaded and preprocessed before being used. Given the number of features at hand and the large size of the dataset, performance was poor while accuracy was low. This low prediction accuracy led to the generation of false alerts, consequently, legitimate alerts used to pass without an action assuming them as false. To deal with large size and false alarms, the proposed voting-based feature reduction approach proved to be highly beneficial in reducing the dataset size by selecting only the features secured majority votes. Outcome collected prior to and following the application of the proposed model were compared. The findings reveal that the proposed approach required less time to predict, at the same time predicted accuracy was higher. The proposed approach will be extremely effective at detecting intrusions in real-time environments and mitigating the cyber-attacks.

KEYWORDS

Efficient intrusion prediction, Feature selection with voting based method, machine learning based intrusion prediction, Minimizing false alarms

INTRODUCTION

The cyber security of IT infrastructure is becoming increasingly important. Computer and network security has become increasingly popular as a result of increased and innovative cyber attacks (CERT, 2018). An adversary, either internal or external to the system, can launch an attack on IT infrastructure. Insider attacks have recently emerged as a major threat to network security (ClearSwift, 2017; HayStax,

DOI: 10.4018/IJDSST.323191

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

2017). Before an attack on a computer or network occurs, adversaries investigate vulnerabilities to exploit the system. Vulnerabilities can exist at any level of the IT infrastructure, including application software, operating systems, and hardware. Vulnerabilities discovered can be exploited to compromise data and render services inaccessible. Despite the fact that network perimeters are well fortified to protect IT resources, adversaries discover novel ways to attack and penetrate networks. Attacks such as Phishing, DDoS, and intrusion are the major cyber-attack contributors (Alomari, Manickam, Gupta, Karuppayah, & Alfariis, 2012; Yu, 2014; Zargar, Joshi, & Tipper, 2013).

Network Vulnerabilities

Vulnerabilities can be defined as a bug or misconfiguration in a software system that is exploited by adversaries to attack a host or a network system (Bazaz & Arthur, 2007; Benton, Camp, & Small, 2013; Bishop & Bailey, 1996). Given the importance of cyber attacks, the Computer Emergency Response Team (CERT) maintains a dedicated portal to notify advanced vulnerabilities and recommend remedial action to plug the vulnerabilities. (CERT, 2018).

An attack on a host or network can be exercised only if a vulnerability exists. Once a vulnerability is exploited, an adversary can intrude into the computer or a network system and is likely to cause huge damage. The attacker exploits the vulnerabilities and succeeds in attacking the target (Wang, Jajodia, Singhal, Cheng, & Noel, 2014). For instance, vulnerabilities in Adobe Flash, and Adobe Acrobat Reader has caused several attacks. To fix the vulnerabilities, Adobe released a series of patches.

Intrusion Detection System

The intrusion detection system is a software that is developed to detect intrusions in a computer or network system. The working principle involves tracing the malicious software demonstrating distinct behaviour relative to the ordinary traffic. Indeed, the need for intrusion prevention was more stressed instead of being limiting to intrusion detection by authors (Cai, Mei, & Zhong, 2018). Ordinary anti-virus software fails to detect such types of advance malicious behaviour (SentinelOne, 2018). Seamless connectivity coupled with accelerated growth in PCs, smartphones, tablets, and internet connectivity offers a great opportunity for adversaries to creep from one device to another (Shelke, Sontakke, & Gawande, 2012; Shakshuki, Kang, & Sheltami, 2013). This leads to a compromise in security and causes the expansion of malicious software. In a network, resources such as nodes or a host can be compromised by intruder on the periphery. Accordingly, IDS can be categorized into: a) host-based b) network-based, and c) periphery-based.

To trace the intrusion, IDS primarily employs either statistical or data mining methods. Usage of data mining techniques is not new in IDS and rigorously employed by authors (Berson & Smith, 1997). In order to combat the new features in malicious software, several new techniques were proposed to thwart intrusion (Aburomman & Reaz, 2016; Altwaijry & Algarny, 2012). To keep IDS usage costs low, authors (Alharkan & Martin, 2012) have proposed a public cloud-based approach to detect intrusion. As a result, the user can gain access to advanced and updated IDS while paying for limited usage.

Present Trend in Security Attacks

To provide the ease of use and flexibility, data maintained on internet is growing. Government initiative and encouragement has further accelerated the growth in digital data. To offer ease of use and flexibility variety of operations related to bank, FinTech, Ecommerce, citizen services, etc can be accessed using internet based accessibility from anywhere using variety of devices. In the same line, financial activities employing technology has grown manifold. Owing to the importance of data, the cost of downtime is higher for a financial institution, and vulnerability is correlated to the number of financial records maintained by the financial institution (Roumani, Nwankpa, & Roumani, 2016). Adversaries are trying hard to develop advanced malware that can penetrate the existing security solutions (Symantec, 2018). Symantec security threat report 2018 outline the current state of security and the emerging areas of attack (Symantec, 2018).

Recently, a new form of malicious software termed ransomware gained momentum and caused huge damage in terms of reputation as well as monetary terms (Brewer, 2016; Newman, 2018; Singh, 2018). Universities, hospitals, and financial institutions across the world are some of the major victims that experienced ransomware. Attackers demanded huge ransom to free the full or partial data (Logan, 2017; Solomon, 2017; Osborne, 2018). Owing to the damage caused and ransom demand, it is also termed as cyber terrorism (Knake, 2010). Attacks have also occurred on Bitcoin, which is considered relatively safe in comparison to other internet-based environments.

Problem Statement

The primary function of IDS is to generate the alert as early as possible based on the severity level of the alert (Chatziadam, Askoxylakis, Petroulakis, & Fragkiadakis, 2014). This allows the administrator to combat the threat by taking appropriate action. Generation of timely and accurate alerts ensures that the administrator is not in a quandary and is not relying on acting experience, but rather on the system to combat the threat. Tree-based algorithms, support vector-based methods, and naive methods are the most widely used machine learning-based methods for effectively classifying target classes (Koc, Mazzuchi, & Sarkani, 2012; Kukreja, Karamchandani, Khandelwal, & Jewani, 2015; Li, et al., 2012; Liao, Lin, Lin, & Tung, 2013).

Research Objectives

Considering the problem at hand, this work has undertaken the following objectives to explore intrusion and improve efficiency to categorize intrusion:

- Identifying important features that have a strong bearing on intrusion.
- Reducing the dataset according to the identified important feature(s) and evaluating the impact on classification.
- Measuring the accuracy and performance gain achieved by using the proposed method before and after dataset reduction.

The rest of the manuscript is organised as follows: Section 2 presents the literature review and the challenges identified based on the review. Section 3 elucidates the research methodology used. Section 4 presents the proposed voting-based model. Finally, section 5 presents the results and discussion of the experiment carried out.

RELATED WORK

Noteworthy contribution related to this work has been presented herein:

To detect the intrusion, a KNN based method was employed by the authors (Wagh, Neelwarna, & Kolhe, 2012), they claimed that the outcome of KNN is more efficient than the earlier existing methods (Wagh, Neelwarna, & Kolhe, 2012). In another work, modified version of SVM that employ the bat algorithm to synchronize the feature selection was proposed by (Chen, Zhichun, Xia, & Liu, 2013). The authors used a support vector-based machine algorithm named (BA-SVM). In another work, authors (Cho, Shon, Choi, & Moon, 2013) suggested the usage of machine learning based algorithm to detect the zero-day attack. In another work, a study related to malware and advance persistent attack (APT) was undertaken by the authors (Kim & Park, 2014). The proposed APT based method predict the intrusion by dividing the dataset into several stages at the time of learning, testing, and finally the evaluation method (Kim & Park, 2014). The authors (Yin, Chen, & Kim, 2014) introduced the method with the name LDFGB that takes the outlier into account to predict and understand the data. Authored claimed it to be more accurate than the one currently existing. The proposed algorithm was capable of covering any type of shape.

The authors (Aleroud & Karabatis, 2017) described the IDS control as a difficult task. In order to enrich the readers; authors claimed to present a new data mining based taxonomy of the previous work carried out. The key objective of the article was to represent the work of data analytics in Intrusion detection systems especially related to contextual information. The authors also highlighted the weakness of the existing techniques and suggested the layered approach for effective prediction of attacks. To deal with the intrusion in distributed environment, an efficient technique that deeply analyzed the component of the intrusion and classified them into four basic units i.e. sensors, a database, an analyzer, and a responsive unit, was proposed by (Platonov & Semenov, 2017). In another work, issue related to false classification that results in raising the false alarm was undertaken by (Shah, Aggarwal, & Chaubey, 2017). The authors proposed the multi-sensor type approach for improving the overall accuracy. The technique was a mathematical model that fuses the evidence and initiated the global decision.

Mobile Adhoc Network (MANET) is ever increasing. MANET allows a system to join a network. Accordingly, security concern is growing in MANET. Wireless network-based intrusion system is ill-suited for MANET. Accordingly, trace back based intrusion detection system was proposed by (Umamaheswari & Kalaavathi, 2018). The proposed application had to reside on a server that can monitor the network based on the weightage system. A mobile-based intrusion detection system was proposed by (Bala, Jothi, & Chandrasekar, 2018). The proposed technique was based on learning with the help of traffic analysis. It was including the time-variant to trace the data traffic.

Findings

Based on the aforementioned literature review, the challenges identified are mentioned herein.

- i. Machine learning-based techniques such as KNN, SVM and decision tree are used to deal with the malicious application. Once, malicious data is traced, an alert notifying the intrusion is generated and subsequently, dealt with by the administrator. Lack of experience or foresightedness results in a poor configuration that is well exploited by the adversaries.
- ii. The complexity of a network is growing owing to the wider connectivity with the devices and inclusion of the Internet of things. The smartphone equipped with an android based operating system is acting as a computer and is widely used for data exchange and connection to the outside world. Such devices can be easily infected by sending malicious software in the form of mail. Apps are frequently downloaded and installed to meet short and long-term goals. During the selection of apps, no detailed study or the supplier's history is examined.
- iii. Infected devices connected with the computer network can easily spread the malware. Adhoc networks and legacy networks will be at risk due to the emergence of android based malware.

The aforementioned contributions have highlighted the need for a technique capable of dealing with the network complexity that has arisen as a result of seamless connectivity with a wide range of devices. Because the data is stored on the server, the approach should be scalable and effective enough to generate an accurate alert in a timely manner. A timeline can be created by devising a noble strategy or by reducing the size of the data at hand.

RESEARCH METHODOLOGY

Data Description and Selection of Dataset

A dataset with the nomenclature UNSW15 (University of New South Wales) was downloaded in order to carry out the experiment. This dataset was created by the IXIA PerfectStorm tool in the Australian Centre of Cyber Security's Cyber Range lab (Moustafa & Slay). To capture 100 GB of raw traffic,

TCPdump is used. Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, reconnaissance, shellcode, and Worms are the nine classes in this dataset. The dataset contains approximately two million and 540,044 records spread across four csv files. There were 49 columns in total. The dataset is available for viewing and downloading at <https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys>.

Other options included data sources such as DARPANET, which was created in 1999-2000 (Dhanabal & Shantharajah, 2015). This dataset is extremely popular among security researchers, and it has been used in a number of studies. This dataset, however, is not used for this work because it is quite old, having been created around 20 years ago. We believe it may have lacked some of the advanced attack features that have evolved over the years.

CAIDA and NALR are two other datasets available for IDS, and they are widely used by researchers all over the world to compare their models to this dataset. However, only statistical methods can be used for CAIDA and NALR. As a result, they were dropped.

Despite the fact that NIST recognised the importance of dataset availability for researchers and stressed the importance of supplying updated datasets on a regular basis, updated data is still lacking despite repeated requests. In this regard, the dataset at our disposal was the best fit. Furthermore, it was saved in 2017 and thus contains advanced trends.

System Configuration

Machine learning techniques supported by the 'R' application are used to carry out the experiment. To that end, several packages were required, such as rpart, earth, rle, and so on. They were all downloaded from the internet. The experimental machine is set up as follows:

- *Operating System: Windows 8.1*
- *RAM: 16 GB*
- *Processor: AMD Fx-4300 Quad Core, 3.80 Ghz*
- *Application Software: R, version 3.5.3*

Sample Drawing Technique

The sample drawing technique is critical to the success of the data analysis technique. The method used to draw the sample should be unbiased while also being time efficient. To achieve the best results, the following sample drawing techniques were used in this work:

- Random Sample
- Hold out method
- Cross-Validation
- Bootstrapping

Feature Selection

The dataset contains 49 attributes. When the algorithm is applied to such a large dataset, performance suffers dramatically. It takes hours to get the result. Such a system cannot be justified in IDS, where the administrator must act immediately upon receiving an alert. Simultaneously, predictions used to be moderate. As a result, we have reduced the dataset by removing features that may be redundant in the dataset. To that end, the algorithms discussed herein were used to select important features:

- Boruta Method
- Decision tree-based method
- Recursive feature elimination (RFE) Method
- Linear Model

Machine Learning Algorithms

Machine learning algorithms are classified as either supervised or unsupervised learning techniques. The model learns in supervised learning techniques by working on the available dataset; once trained, the model is applied to testing data. Groups are initially unknown in the unsupervised method. Nave Bayes, SVM, decision tree, forest tree, KNN, and other popular machine learning algorithms falls under supervised learning. We tested our model on the following algorithms, in addition to the ones mentioned above.

- Naïve Bayes
- SVM
- Decision Tree

Reasons to select the aforementioned algorithms are enumerated herein:

- In cases where history is available, the supervised-based method is well suited. Since we had access to the training set, we used supervised learning classification techniques rather than unsupervised ones.
- The need for scaling was the second reason. The aforementioned algorithms do not require data scaling, whereas other algorithms, such as neural networks and KNN, do.
- Scaling is incompatible with the need for real-time analysis, especially in large datasets, because scaling requires additional time. Furthermore, they are working on raw data, whereas other popular techniques, such as KNN and NN, can be used on normalised data. Our goal was to investigate the performance and efficiency of undertaken algorithms, particularly those that operate on raw data rather than transformed data.

PROPOSED VOTING BASED METHOD

In a multi-dimension dataset, not all attributes are required; instead, the model uses some of them. As a result, we identified the most appropriate features in the IDS dataset to improve the performance of three classification algorithms used to generate the early warning system (Chatziadam, Askoxylakis, Petroulakis, & Fragkiadakis, 2014).

This work begins with identifying the characteristics that govern the target class's behaviour. To that end, we used popular feature selection techniques such as RLE, decision tree-based models, and linear models to identify important features. Finally, variables were chosen based on vote count. A variable must receive at least 50% of the total votes cast in order to be chosen. That is, if we used all five techniques, the total number of votes is 5. A variable must have been voted on by three or more techniques in order to be chosen.

Model Accuracy Using Confusion Matrix

This study relies on a confusion matrix, which is a matrix-based summary of the target class, to predict model accuracy. This method compares the previously known classes to the ones predicted by the proposed model. As a result, it divides the prediction into four categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN) (FN). It can be understood with the following example.

Meanings of the terms used are as follows:

TN: Class is negative and predicted as negative.

FN: Class is negative and predicted as positive.

FP: Class is positive and predicted as negative.

TP: Class is positive and predicted as positive.

Table 1. Confusion Matrix

	Predicted 'No'	Predicted 'Yes'
Actual 'No'	(TN)	(FN)
Actual 'Yes'	(FP)	(TP)

Thereby accuracy of the model can be computed by employing the laid down formulae.
 Accuracy=predicted correct class/ Total prediction
 i.e.

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN}$$

Target classes that are accurately predicted are assigned the value of '1' and '0' otherwise. To be more generalized, values falling in the diagonal are true and the rest of the values are treated as false.

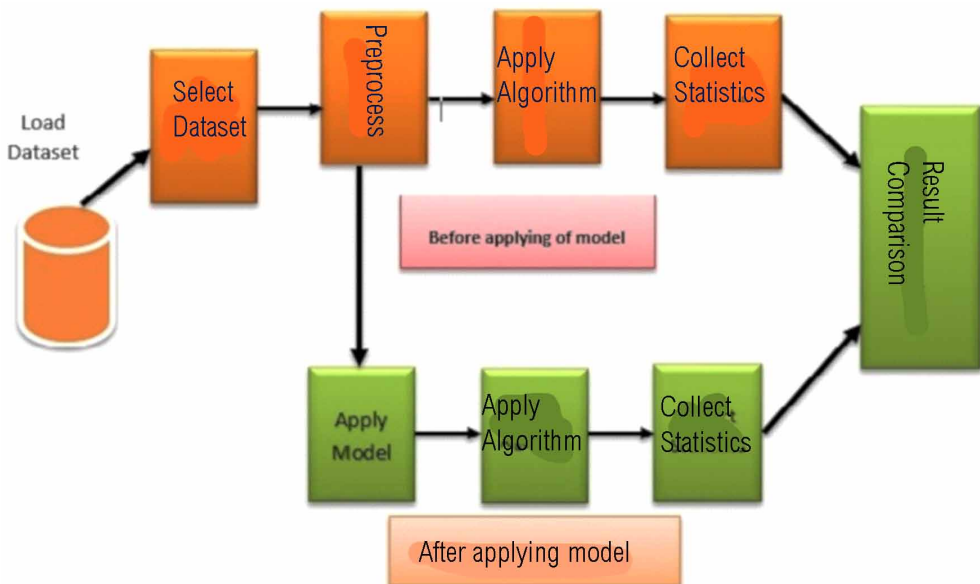
Working of Proposed Model

After preliminary requirement, it follows one of the two paths available. The first path represents before applying the proposed model, and the other one after applying the proposed model. Each path comprises four phases that include, applying a machine learning algorithm, output collection, and result discussion. Phases of the proposed system have been elucidated herein:

Selecting and Loading Dataset

The model begins with loading the target dataset. During this phase, the entire dataset is loaded in memory. Once the dataset is loaded, other phases follow.

Figure 1. Working of proposed model



Preprocessing Phase

Once, the dataset is loaded, preprocessing is applied to clean the data. The complete dataset was not used for this research, instead, 2000 instances relating to attack category and the same quantity of non-attack category were selected. During the instance selection, all the instances were offered equal chance of selection; therefore, instances were selected on a random basis. This phase was common to both the paths i.e. before applying the proposed model and after applying the proposed model.

Model Application

To work on the dataset, we have identified the important features in our dataset. To this end, prominent feature selection algorithms including Boruta, RFE, Linear Model, Decision Tree, and GLM were used. The outcome of feature selection varied from one algorithm to another (Cheng, Bao, & Bao, 2016). Since each algorithm proposed its own set of features thereby we have gathered the outcome of each method. To arrive at a concrete decision, we have proposed voting based method that selects the features based on the majority votes.

Data Collection Prior to Model Application

Under this phase, classification-based methods that include Naïve Bayes, Tree-based, and SVM were applied. The outcome of the aforementioned methods was collected for subsequent phase.

Applying the Proposed Model

Once the dataset is loaded and preprocessing is over, we applied our model and reduced the dataset size. Classification-based algorithms were applied on the reduced datasets to enable them to learn and predict.

Result Comparison Phase

Once the experiment is conducted, results were collected without applying the proposed model and after applying the proposed model and results were compared.

Proposed Algorithm Pseudocode

Pseudo code of the algorithm applied has been given herein:

Psuedo code

```
loadDataset          #Load the dataset
Load Library         #Load related 'r' Library
ApplyFeatureSelectionAlgorithm # Apply Feature Selection
Algorithm
SelectImportantFeatures # Select Important Feature
AssignVote          # Assign the vote to
selected variable
CountVote           # Count the vote of
variable
DecideWinner        # Decide the winner
ReduceDataSetAccordingToImportantFeature #Reduce dataset
by removing redundant features
ApplyAlgorithm      # Apply the Algorithm
CollectData         # Collect data
ComputeAccuracy     # Compute Accuracy of the model
ComputePerformance  # Compute performance in terms of time
CompareResult       #Compare results before and after
the application of the proposed model
```


Table 2. Feature votes assignment and selection based on votes

Features	Boruta	Rfe	LM	Random forest	GLM	Result
Dur	1	0	0	0	0	NS
Sbytes	1	0	1	0	0	NS
Dbytes	1	1	1	0	0	Selected
Sttl	1	1	1	1	1	Selected
Dttl	1	1	1	1	1	Selected
Sload	1	0	1	0	0	NS
Dload	1	1	1	0	1	Selected
Spkts	1	0	1	0	1	Selected
Dpkts	1	1	1	0	0	Selected
Swin	1	0	1	0	0	NS
Dwin	1	0	0	0	0	NS
Stcpb	1	0	0	0	0	NS
dtcpb	1	0	0	0	0	NS
trans_depth	1	0	0	0	0	NS
res_bdy_len	1	0	1	0	0	NS
Sjit	1	1	1	0	0	Selected
Djit	1	0	1	0	0	NS
Sintpkt	1	0	1	0	0	NS
Dintpkt	1	0	1	0	0	NS
attack_cat	1	0	0	0	0	NS

RESULTS AND DISCUSSION

Assignment of Votes

After applying feature selection algorithms such as Boruta, RFE, LM, Random Forest, and GLM, feature selection differs from one algorithm to another. Table 2 presents the results of the feature selection method. The table indicates that the features chosen by each method vary significantly, and in order to reach a consensus, there should be a method for selecting the most reliable features.

As a result, we have introduced the voting-based method to select the variables with the greatest presence. To that end, a matrix is created that records the votes based on the selection. If the feature is selected, a '1' vote is assigned; otherwise, a '0' vote is assigned. Following that, votes for each feature are counted. When the feature receives '3' or more votes, the winner is declared.

To count the votes for a row, the method employed

$$RS_{i n} = \sum_{i=0}^r \sum_{j=0}^{n-1} R_i C_j$$

Where RS is the row sum

'R' represents row.

'n' represents the last column of a row.

'i' represents the 'row number.
'j' represents the column number.
'C' represents column.

As a result, votes are counted and assigned to the appropriate features. A feature with fifty percent or more votes is considered a winner and will be levelled as such. In the final dataset, only the features that were victorious were kept, while the rest were removed. Table 2 depicts the selected features in the undertaken dataset, which include dbytes, sttl, dttl, Dload, Spkts, Dpkts, and Sjit.

Dataset Reduction

In our dataset, all the features are not important instead they are part of the data and increasing the size of the dataset. Due to the larger size, identifying whether the request is containing anything malicious needs more time at the same time complexity of the data grows. Thereby, dataset is reduced by including only those fields that are needed. Eventually, our dataset includes the feature that has won under the feature selection method.

Performance Comparison

Our primary focus in the performance comparison was latency. We computed the time taken by each algorithm before and after applying the proposed model. Figure 2 depicts the outcome of both.

Figure 2 shows that all three algorithms have significantly improved in performance. However, the highest gain was observed in the case of SVM.

Accuracy Comparison

We used a confusion matrix to compare accuracy. Figure 3 depicts the results. According to the graph, our model's accuracy has also improved. Again, the major improvement is seen in SVM and the least in tree-based models.

Figure 2. Outcome performance comparison

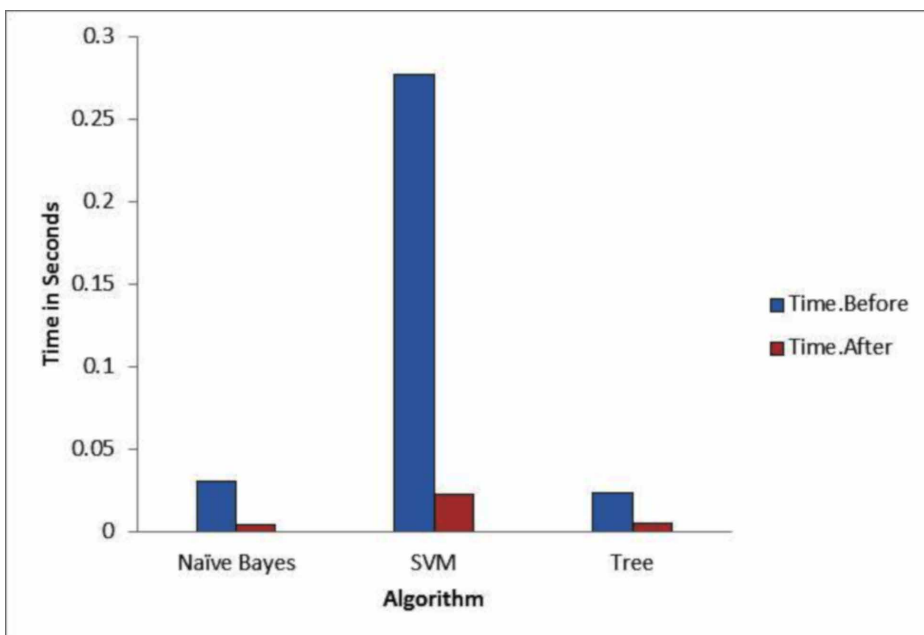
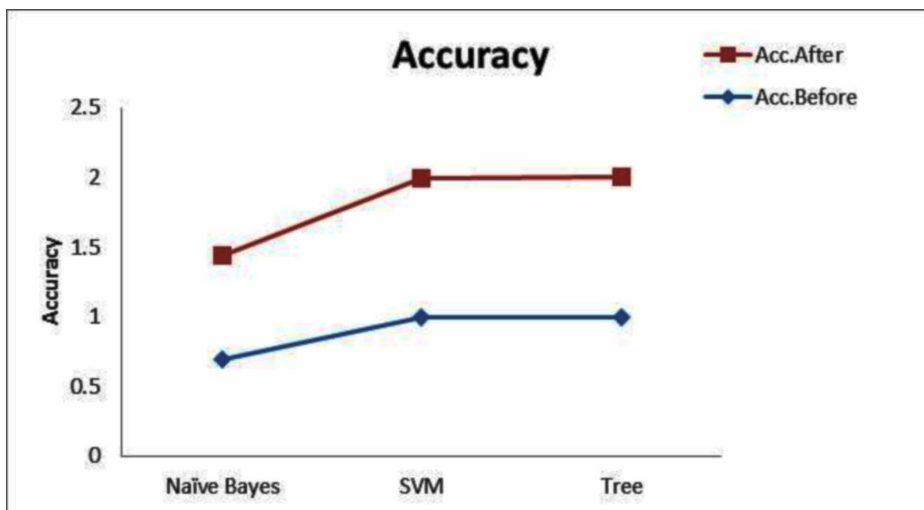


Figure 3. Accuracy comparison before and after applying the model



CONCLUSION

An intrusion detection system is a solution required to detect and prevent intrusions. Despite extensive research in this area, a reliable method for detecting attacks with greater accuracy and reliability is lacking. In a real-time environment, it is difficult for any intrusion detection system to detect a wide range of attacks with low false alarms. This proposed method proved to be efficient enough to handle large amounts of data while remaining performant in the deployed environment. Techniques such as Nave Bayes, decision tree based, and SVM models were used to determine the factors that govern the behaviour of the target class.

We chose the variables based on the voting mechanism to make our decision. At the same time, a time efficiency gain distinguishes it from all other existing approaches. As a result, the proposed method is well suited for use in real time. During the experimentation phase, the proposed method undoubtedly produced promising results.

Gains in accuracy and time support its implementation in an operational environment. However, due to a lack of operational infrastructure, the proposed method has not been tested in a real-world setting. Furthermore, at the time of the experiment, it had not been integrated with intelligent applications such as intrusion detection. The performance of the proposed method on the operational server with traffic comparable to that seen in the operational environment remains to be seen. Once successful in simulated environment, proposed voting-based methods can be included with intrusion detection system for promising outcome.

COMPLIANCE WITH ETHICAL STANDARDS

There is no conflict of interest of any of the author(s). This article does not contain any studies with human participants performed by any of the authors.

REFERENCES

- Aburomman, A. A., & Reaz, M. B. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, 38, 360–372. doi:10.1016/j.asoc.2015.10.011
- Aleroud, A., & Karabatis, G. (2017). Contextual information fusion for intrusion detection: A survey and taxonomy. *Knowledge and Information Systems*, 52(3), 563–619. doi:10.1007/s10115-017-1027-3
- Alharkan, T., & Martin, P. (2012). IDSaaS: Intrusion detection system as a service in public clouds. *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, (pp. 686-687). IEEE. doi:10.1109/CCGrid.2012.81
- Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfari, R. (2012). Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art. *arXiv preprint arXiv:1208.0403*.
- Altwaijry, H., & Algarny, S. (2012). Bayesian based intrusion detection system. *Journal of King Saud University-Computer and Information Sciences*, 24(1), 1–6. doi:10.1016/j.jksuci.2011.10.001
- Andrea, I., Chrysostomou, C., & Hadjichristofi, G. (2015). Internet of Things: Security vulnerabilities and challenges. *Computers and Communication (ISCC), 2015 IEEE Symposium on*, (pp. 180-187). IEEE.
- Bala, K., Jothi, S., & Chandrasekar, A. (2018). An enhanced intrusion detection system for mobile ad-hoc network based on traffic analysis. *Cluster Computing*, 1–8.
- Bazaz, A., & Arthur, J. D. (2007). Towards a taxonomy of vulnerabilities. *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, (pp. 163a--163a).
- Benton, K., Camp, L. J., & Small, C. (2013). OpenFlow vulnerability assessment. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, (pp. 151-152). ACM. doi:10.1145/2491185.2491222
- Berson, A., & Smith, S. J. (1997). *Data warehousing, data mining, and OLAP*. McGraw-Hill, Inc.
- Bishop, M., & Bailey, D. (1996). *A critical analysis of vulnerability taxonomies*. Tech. rep., California Univ Davis Dept Of Computer Science.
- Bisht, P., Hinrichs, T., & Venkatakrishnan, V. N. (2015, 8). *System and a method for automatically detecting security vulnerabilities in client-server applications*. Google Patents.
- Brewer, R. (2016). Ransomware attacks: Detection, prevention and cure. *Network Security*, 2016(9), 5–9. doi:10.1016/S1353-4858(16)30086-1
- Cai, C., Mei, S., & Zhong, W. (2018). Configuration of intrusion prevention systems based on a legal user: The case for using intrusion prevention systems instead of intrusion detection systems. *Information Technology Management*, 1–17.
- CERT. (2018, March 15). *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sector*. US-CERT. <https://www.us-cert.gov/ncas/alerts/TA18-074A>
- Chatziadam, P., Askoxylakis, I. G., Petroulakis, N. E., & Fragkiadakis, A. G. (2014). Early warning intrusion detection system. *International Conference on Trust and Trustworthy Computing*, (pp. 222-223). Springer. doi:10.1007/978-3-319-08593-7_22
- Chen, Y., Zhichun, L. I., Xia, G., & Liu, B. (2013, 8). *Matching with a large vulnerability signature ruleset for high performance network defense*. Google Patents.
- Cheng, C., Bao, L., & Bao, C. (2016). Network Intrusion Detection with Bat Algorithm for Synchronization of Feature Selection and Support Vector Machines. *International Symposium on Neural Networks*, (pp. 401-408). Springer. doi:10.1007/978-3-319-40663-3_46
- Cho, J., Shon, T., Choi, K., & Moon, J. (2013). Dynamic learning model update of hybrid-classifiers for intrusion detection. *The Journal of Supercomputing*, 64(2), 522–526. doi:10.1007/s11227-011-0698-x
- ClearSwift. (2017). *The Enemy Within research*. Clearswift.com: <https://www.clearswift.com/sites/default/files/images/blog/enemy-within.pdf>

- 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, 446–452.
- Elayidom, M. S. (2015). *Data Mining and business intelligence*. Delhi: Cengage learning India.
- Goseva-Popstojanova, K., & Perhinschi, A. (2015). On the capability of static code analysis to detect security vulnerabilities. *Information and Software Technology*, 68, 18–33. doi:10.1016/j.infsof.2015.08.002
- HayStax. (2017). *Insider Attacks 2017 Insider Threat Study*. HayStax. <https://haystax.com/blog/whitepapers/insider-attacks-industry-survey/>
- Jimenez, M., Le Traon, Y., & Papadakis, M. (2018). Enabling the Continuous Analysis of Security Vulnerabilities with VulData7. *IEEE International Working Conference on Source Code Analysis and Manipulation*. IEEE.
- Jones, R. R., McCaffrey, K. J., Clegg, P., Wilson, R. W., Holliman, N. S., Holdsworth, R. E., & Waggott, S. (2009). Integration of regional to outcrop digital data: 3D visualisation of multi-scale geological models. *Computers & Geosciences*, 35, 4–18.
- Kannan, S., Karimi, N., Sinanoglu, O., & Karri, R. (2015). Security vulnerabilities of emerging nonvolatile main memories and countermeasures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(1), 2–15. doi:10.1109/TCAD.2014.2369741
- Kelekar, S. G. (2012). *Systems and methods for real-time network-based vulnerability assessment*. Google Patents.
- Kim, Y.-H., & Park, W. H. (2014). A study on cyber threat prediction based on intrusion detection event for APT attack detection. *Multimedia Tools and Applications*, 71(2), 685–698. doi:10.1007/s11042-012-1275-x
- Knake, R. K. (2010, Feb 12). *Cyberterrorism Hype v. Fact*. Retrieved from Council on Foreign Relations. <https://www.cfr.org/expert-brief/cyberterrorism-hype-v-fact>
- Koc, L., Mazzuchi, T. A., & Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications*, 39(18), 13492–13500. doi:10.1016/j.eswa.2012.07.009
- Kreutz, D., Ramos, F., & Verissimo, P. (2013). Towards secure and dependable software-defined networks. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, (pp. 55–60). ACM. doi:10.1145/2491185.2491199
- Kukreja, K., Karamchandani, Y., Khandelwal, N., & Jewani, K. (2015). *Intrusion Detection System*. International Journal of Scientific and Research Publications.
- Kustarz, C., Huston III, L. B., Simpson, J. A., Winquist, J. E., Barnes, O. P., & Jackson, E. (2016, 8). *System and method for denial of service attack mitigation using cloud services*. Google Patents.
- Lantern, M. (2018, Apr). *The value of 20/20 hindsight in cybersecurity*. CSO IDG. <https://www.csoonline.com/article/3268285/data-protection/the-value-of-20-20-hindsight-in-cybersecurity.html>
- Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications*, 39(1), 424–430. doi:10.1016/j.eswa.2011.07.032
- Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1), 16–24. doi:10.1016/j.jnca.2012.09.004
- Lin, W.-C., Ke, S.-W., & Tsai, C.-F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, 78, 13–21. doi:10.1016/j.knosys.2015.01.009
- Logan, S. (2017). *Lessons learned at University of Calgary as global ransomware attack foiled*. Calgary Herald. <https://calgaryherald.com/news/local-news/lessons-learned-at-university-of-calgary-as-global-ransomware-attack-foiled>
- Manky, D. (2010, Nov). *Top 10 vulnerabilities inside the network*. NetworkWorld. <https://www.networkworld.com/article/2193965/tech-primers/top-10-vulnerabilities-inside-the-network.html>

- McClure, S. C., Kurtz, G., Keir, R., Beddoe, M. A., Morton, M. J., Prorise, C. M., & Abad, C. (2012, 3). *System and method for network vulnerability detection and reporting*. Google Patents.
- Mitchell, R., & Chen, R. (2013). On survivability of mobile cyber physical systems with intrusion detection. *Wireless Personal Communications*, 68(4), 1377–1391. doi:10.1007/s11277-012-0528-3
- Moustafa, N., & Slay, J. (2018.). *The UNSW-NB15 Dataset Description*. UNSW. <https://www.unsw.adfa.edu.au/uns-w-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>
- Newman, L. H. (2018, april). Atlanta Spent \$2.6M to Recover From a \$52,000 Ransomware Scar. *Wired*. <https://www.wired.com>
- Nicodemus, B., & Stephens, B. E. (2015, 2). *Methods and systems for controlling access to computing resources based on known security vulnerabilities*. Google Patents.
- Nunes, P., Medeiros, I., Fonseca, J., Neves, N., Correia, M., & Vieira, M. (2018). An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios. *Computing*, ●●●, 1–25.
- Om, H., & Kundu, A. (2012). A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*, (pp. 131-136).
- Osborne, C. (2018). *US hospital pays \$55,000 to hackers after ransomware attack*. ZD Net. <https://www.zdnet.com>: <https://www.zdnet.com/article/us-hospital-pays-55000-to-ransomware-operators/>
- Ou, X., Govindavajhala, S., & Appel, A. W. (2005). MulVAL: A Logic-based Network Security Analyzer. *USENIX Security Symposium*, 8.
- Pan, S., Morris, T., & Adhikari, U. (2015). Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, 6(6), 3104–3113. doi:10.1109/TSG.2015.2409775
- Patel, A., Taghavi, M., Bakhtiyari, K., & Junior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, 36(1), 25–41. doi:10.1016/j.jnca.2012.08.007
- Pfleeger, C. P., & Pfleeger, S. L. (2012). *Analyzing computer security: a threat/vulnerability/countermeasure approach*. Prentice Hall Professional.
- Piessens, F., & Verbauwhede, I. (2016). Software security: Vulnerabilities and countermeasures for two attacker models. *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*, (pp. 990-999). doi:10.3850/9783981537079_0999
- Pitilakis, K., Argyroudis, S., Kakderi, K., & Selva, J. (2016). Systemic vulnerability and risk assessment of transportation systems under natural hazards towards more resilient and robust infrastructures. *Transportation Research Procedia*, 14, 1335–1344. doi:10.1016/j.trpro.2016.05.206
- Platonov, V. V., & Semenov, P. O. (2017). An adaptive model of a distributed intrusion detection system. *Automatic Control and Computer Sciences*, 51(8), 894–898. doi:10.3103/S0146411617080168
- Ritchey, R. W., & Ammann, P. (2000). Using model checking to analyze network vulnerabilities. *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, (pp. 156-165). IEEE.
- Roumani, Y., Nwankpa, J. K., & Roumani, Y. F. (2016). Examining the relationship between firm_s financial records and security vulnerabilities. *International Journal of Information Management*, 36(6), 987–994. doi:10.1016/j.ijinfomgt.2016.05.016
- Sabahi, F., & Movaghar, A. (2008). Intrusion detection: A survey. *Systems and Networks Communications, 2008. ICSCN'08. 3rd International Conference on*, (pp. 23-26).
- Sadeghi, A., Bagheri, H., & Malek, S. (2015). Analysis of android inter-app security vulnerabilities using COVERT. *Proceedings of the 37th International Conference on Software Engineering-Volume 2*, (pp. 725-728). IEEE. doi:10.1109/ICSE.2015.233
- Senie, D., & Ferguson, P. (1998). Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. *Network (Bristol, England)*.

- SentinelOne. (2018, March). *Survey: 53 Percent Of Organizations Blame Legacy Antivirus Protection For Failed Ransomware Prevention*. SentinelOne. <https://www.sentinelone.com/press/survey-53-percent-organizations-blame-legacy-antivirus-protection-failed-ransomware-prevention/>
- Shah, V., Aggarwal, A. K., & Chaubey, N. (2017). Performance improvement of intrusion detection with fusion of multiple sensors. *Complex & Intelligent Systems*, 3, 33-39.
- Shakshuki, E. M., Kang, N., & Sheltami, T. R. (2013). EAACK_a secure intrusion-detection system for MANETs. *IEEE Transactions on Industrial Electronics*, 60(3), 1089–1098. doi:10.1109/TIE.2012.2196010
- Shelke, M. P., Sontakke, M. S., & Gawande, A. D. (2012). Intrusion detection system for cloud computing. *International Journal of Scientific & Technology Research*, 1, 67-71.
- Singh, J. (2018, Jun). *Hidden Cobra Continue to Spread*. VVeGurukul.net. <http://vvegurukul.net/blog/hiddenCobra.php>
- Solomon, H. (2017). *Canadian firm pays \$425,000 to recover from ransomware attack*. IT World Canada: <https://www.itworldcanada.com/article/canadian-firm-pays-425000-to-recover-from-ransomware-attack/394844>
- Symantec. (2018). *2018 Internet Security threat report, Vol 23*. CA: Symantec corporation.
- Todd, M., Koster, S. R., & Wong, P. C. (2016, 2). *System and method for securing a network from zero-day vulnerability exploits*. Google Patents.
- Umamaheswari, A., & Kalaavathi, B. (2018). Honeypot TB-IDS: Trace back model based intrusion detection system using knowledge based honeypot construction model. *Cluster Computing*, 1–8.
- Wagh, S., Neelwarna, G., & Kolhe, S. (2012). A Comprehensive Analysis and Study in Intrusion Detection System Using k-NN Algorithm. *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, (pp. 143-154). IEEE. doi:10.1007/978-3-642-35455-7_14
- Wang, L., Jajodia, S., Singhal, A., Cheng, P., & Noel, S. (2014). k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*, 11(1), 30–44. doi:10.1109/TDSC.2013.24
- Wilton, S., Sedat, B. D., Irizarry, A., Borohovski, M., & Braun, A. K. (2018, 9). *Determining Security Vulnerabilities in Application Programming Interfaces*. Google Patents.
- Yin, S.-n., Chen, Z.-g., & Kim, S.-R. (2014). LDFGB algorithm for anomaly intrusion detection. *Information and Communication Technology-EurAsia Conference*, (pp. 396-404). IEEE. doi:10.1007/978-3-642-55032-4_39
- Yoo, H., & Shon, T. (2016). Challenges and research directions for heterogeneous cyber—physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture. *Future Generation Computer Systems*, 61, 128–136. doi:10.1016/j.future.2015.09.026
- Yu, S. (2014). *Distributed denial of service attack and defense*. Springer New York. doi:10.1007/978-1-4614-9491-1
- Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials*, 15, 2046-2069. IEEE.

Jitendra Singh pursued PhD in the area of cloud computing in 2013. He has qualified the prestigious UGC-NET examination conducted by the UGC of India in the year 2006. With over 16 years of experience in teaching, research and administration, currently he is working with a college of University of Delhi. In addition to the Indian University, he has also contributed as faculty member with the Stratford University, USA, India Campus, as a faculty for over five and half years. Beyond, he has contributed more than two dozen of research articles in the area of cloud computing, security, machine learning. Several of them have been published in reputed journals indexed in Scopus, Inspec, DBLP etc. Besides, he has authored three books namely 'Cloud computing for beginner to researcher', 'Data structure simplified: Implementation with c++', 'Python: Principles and practice'.

Vikas Kumar received M Sc. in Electronics from Kurukshetra University, Haryana, India. This was followed by M Sc in Computer Science and further Ph. D. from the same university. He is a life member of Indian Science Congress Association, Computer Society of India, IETE, ICEIE, IPA, VEDA, IVS and Magnetic Society of India. Dr. Kumar has designed and conducted number of training programs for the corporate sector and is a trainer for a number of Govt of India departments. Along with 15 books, He has more than 100 research papers to his credit in various national and international conferences and journals. Out of which, 55 are with the scopus indexed international journals. He was the Editor of International Quarterly Refereed Journal "Asia-Pacific Business Review" during June 2007-June 2009. Dr. Kumar is presently serving at the Central University of Haryana, Mahendergarh, India and is a visiting Professor at the Indian Institute of Management, Indore and University of Northern Iowa, USA.

Saurabh Shrivastava is working as a professor in institute of basic science (department of mathematical science and computer applications), Bundelkhand University. He has twenty four years of experience in teaching and research. Dr. Saurabh Shrivastava has authored more than thirty research papers in various International journals of repute and presented works at many national and international conferences. Six students have successfully completed Ph.D. under his guidance.