Project report: Machine Learning and Deep Learning Techniques in Network Intrusion Detection

Aakash Chowdhury

Alpen-Adria-Universität, Klagenfurt

Abstract. In the subject of cybersecurity, Anomaly-based Intrusion Detection Systems(AIDS) is a vital and rapidly evolving study area developed to identify different kinds of malicious network traffic and computer usage, which cannot be identified by traditional farewell. In this project, a comparison analysis has been conducted on some popular machine learning and deep learning techniques-based anomaly detectors on a dataset which consists of a wide range of intrusions replicated in a military network environment.

1 Motivation

Advancement in shared networks and internet usage demands increases attention on information system security, particularly on intrusion detection. Due to evolution of several malwares it is quiet difficult to design an intrusion detection system(IDS) with a significant high accuracy rate. According to [Khraisat et al., 2019], malicious attacks have become more complex, and the most difficult task is identifying unknown and disguised malicious software, since malware developers employ various evasion strategies to avoid detection by an IDS.
As stated in [Co, 2002], an intrusion attempt or a threat is a deliberate and unauthorized attempt to (i) access information, (ii) manipulate information, or (iii) render a system unreliable or unusable. For example, activities that would make the computer services unresponsive to legitimate users are considered an intrusion. An IDS is a software or hardware device that detects harmful activity on computer systems in order to preserve system security [Liao et al., 2013].
One commonly used strategy for detection is through anomaly detection, with the explicit assumption that any malicious behavior is anomalous [Arul et al., 2013]. In AIDS, A normal model of a computer system's behavior is built in AIDS utilizing machine learning, statistical-based, or knowledge-based methods. An anomaly is regarded as a significant difference between observed behavior and the model, which can be viewed as an intrusion. This set of approaches is based on the notion that harmful conduct varies from normal user behavior. Intrusions are anomalous user behaviors that differ from usual behavior. The training phase and the testing phase are the two stages in the development of AIDS. The normal traffic profile is utilized in the training phase to build a model of usual behavior, and a new data set is used in the testing phase to determine the system's ability

to generalize to previously unseen intrusions. AIDS can be classified into a number of categories based on the method used for training, for instance, statistically based, knowledge-based, and machine learning-based [Khraisat et al., 2019].
An ideal intrusion detection system is one that has a high attack detection rate along with a 0% false positive rate (the rate of misclassified normal behavior). To find such ideal AIDS, we need to conduct a comparative study of different anomaly detection methods. In this project, we have implemented different machine learning and deep learning-based classifiers, namely Logistic regression with gradient descent algorithm, support vector machine(SVM), K-nearest neighbor(KNN) algorithm and multilayer artificial neural network(ANN). We have considered a data which contains a large variety of intrusions simulated in a military network environment for this comparative experiment. In our experiment, we have observed that KNN and ANN performs significantly better (98-99% accuracy) than the other two methods. We will make a tabulated presentation of the outcomes in the later section.

2 Data

2.1 Data description

The data set to be audited was provided which consists of a wide variety of intrusions simulated in a military network environment. It created an environment to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was focused like a real environment and blasted with multiple attacks. A connection is a sequence of TCP packets starting and ending at some time duration between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Also, each connection is labeled as either normal or as an attack with exactly one specific attack type. Each connection record consists of about 100 bytes. We have collected the data from the kaggle page:
https://www.kaggle.com/sampadab17/network-intrusion-detection
For each TCP/IP connection, 41 quantitative and qualitative features are obtained from normal and attack data (3 qualitative and 38 quantitative features). The class variable(decision vector) has two categories: normal and anomalous. All total 25192 number of connections have been considered in the train dataset, among which 53% are normal and 47% are anomalies.

2.2 Preliminary data processing

For an appropriate implementation of deep learning and machine learning technique we have to do some preprocessing and data manipulation. In our case, we have considered a few which are given as follows:

– **Encoding categorical features:** It was found that the data type of the train dataset that 4 variables(3 features and 1 decision vector) are of object type. So, our first task is to convert them into numeric so that they can be considered for machine learning or deep learning model inputs.

– **Standardize the data:** Standardization of data sets is a common requirement for many machine learning estimators implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance.
– **Splitting Train data:** In our case, test data does not have the the column of decision variable(class variable). Therefore, it is necessary to split our train data for model validation/performance checking purposes. For my implementation, I have considered 2:1 ratio for train$(67\%, 16878)$ and test$(33\%, 8314)$ respectively.
– **Feature Selection:** In IDS data sets, many features are redundant or less influential in separating data points into correct classes. Therefore, features selection should be considered in some machine learning techniques, for example Support Vector Machine(SVM) training.

3 Theoretical Background

Intrusion detection can be thought of as a classification problem where each audit record can be classified into one of a discrete set of possible categories, normal or a particular kind of intrusion. Intrusion detection using data mining have attracted more and more interests in recent years. As an important application of data mining, they aim to meliorate the great burden of analyzing huge volumes of audit data and realizing performance optimization of detection rules.

3.1 Classification techniques for intrusion detection

After having a brief description about data and the type of the decision variable, we can now state that our objectivity is to classify anomaly from a set of individuals so that it would be possible to produce a predicted(estimated) outcome of the class variable which have classes anomaly and normal. So, our learning task is now reduced to produce an appropriate binary classifier from the train data and validate it with the test data [Meza et al.,]. In this section, a brief explanation has been provided about the theoretical ideas behind different classification methods that we have taken into account in our experiment. we will also mention a brief outline about the performance measures that we have considered to validate our methods.

Multi-feature Logistic Regression: Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function. Logistic regression is implemented in Logistic Regression. This implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional l_1, l_2 or Elastic-Net regularization [Schmidt et al., 2013] [Defazio et al., 2014]. As

an optimization problem, binary class l_2 penalized logistic regression minimizes the following cost function:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + C)) + 1)$$

where, y_i is the decision variable, X_i is the matrix of features and w = weight of the model.

Support Vector Machine: Support Vector Machine(SVM) is a discriminative classifier defined by splitting hyper-plane. SVMs use a kernal function to map the training data into a higher dimensioned space so that the intrusion is linearly classified [Chen et al., 2005].

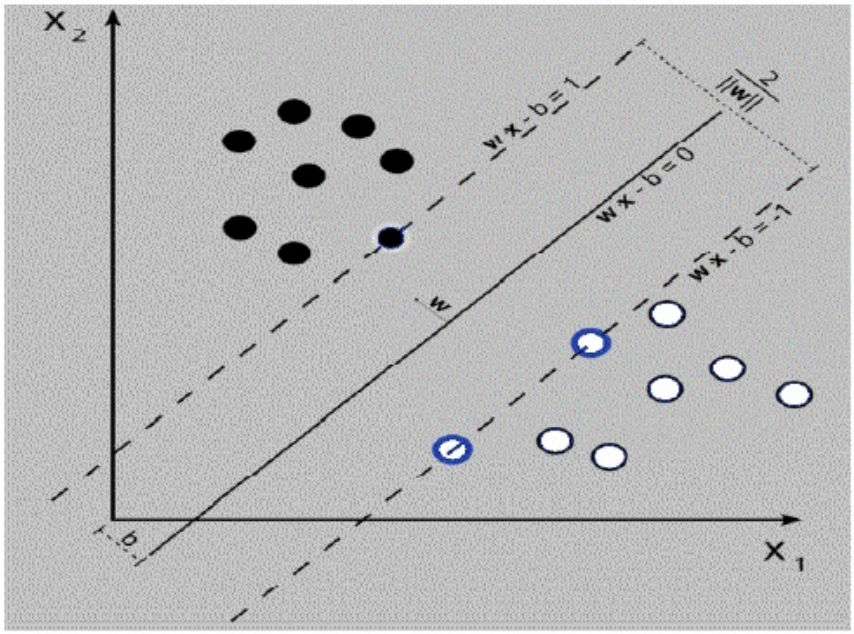
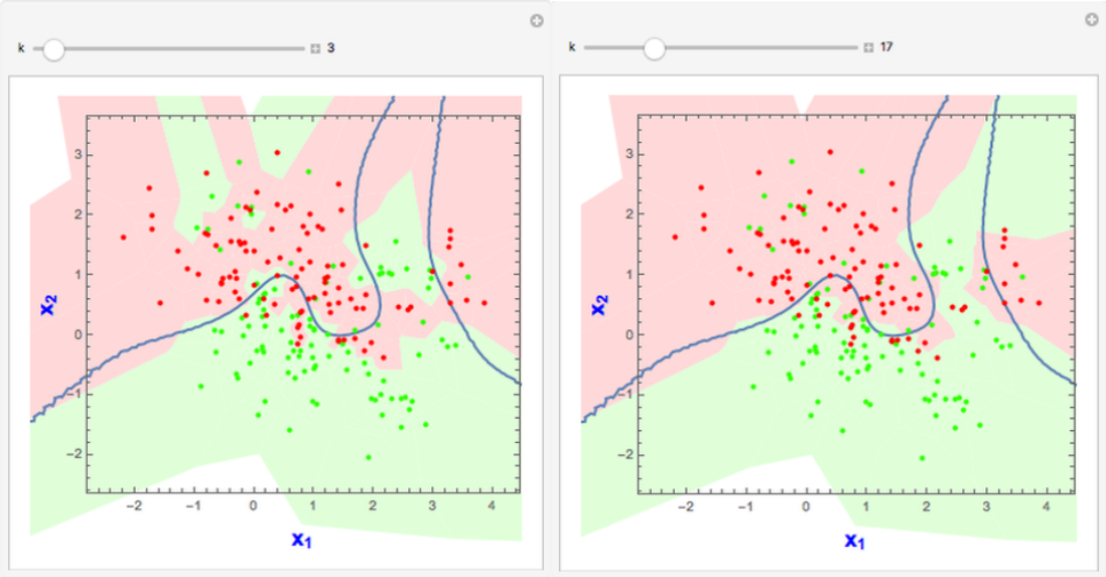


Fig. 1: An example of SVM implementation [Arul et al., 2013]

***K*-nearest neighbor algorithm** The K- Nearest Neighbor(KNN) technique is a typical non-parametric classifier applied in machine learning. The idea of these technique is to name an unlabeled data sample to the class of its K nearest neighbors. To give a sense of its application, we have considered two plots to understand how K-NN method can be used as a binary classifier for a bivariate data.



(a) KNN with $K = 3$ (b) KNN with $K = 17$

Fig. 2: Implementation K-NN method on a bivariate dataset [McLeod, 2011]

Multilayer Artificial Neural Network: ANN is one of the most broadly applied machine-learning methods and has been shown to be successful in detecting different malware. The most frequent learning technique employed for supervised learning is backpropagation (BP) algorithm. The BP algorithm assesses the gradient of the networks error with respect to its modifiable weights. The strength of ANN is that, with one or more hidden layers, it is able to produce highly nonlinear models which capture complex relationships between input attributes and classification labels [Khraisat et al., 2019]. That is the reason multi-layer ANN is one of popular techniques that the researchers are implementing in IDS.

3.2 Validation measures

To measure the performances of different Machine learning and deep learning methods 4 measures are applied in our experiment:

- **Log loss or Cross-entropy loss [Bishop, 2006]:**

$$L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p))$$

  where, y is the decision/class variable corresponding to test data set and p is the predicted probability obtained after training the train data set.
- **Accuracy measure:**

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \mathbb{1}(\hat{y}_i = y_i)$$

where, n_{samples} is the length of the test data set. y and $\hat{y}$ is the original and predicted outcome of the class variable.

– **Confusion matrix:** For visualization, confusion matrix is also a suitable for validation of a single process. Here, True Positive (TP): IDS correctly identify when an legitimate attack occurs.

Table. Confusion Matrix

Confusion Matrix		Predicted Class	
		Attack	**Normal**
Actual Class	Attack	TP	FN
	Normal	FP	TN

Fig. 3: Confusion matrix [Arul et al., 2013]

False Positive (FP): IDS can't identify when an legitimate attack occurs. So it put a false alarm.
True Negative (TN): IDS correctly identify Normal behavior.
False Negative (FN): IDS identify as a normal activity when actually it is an attack.
In our comparison experiment, we have considered False Positive Rate(FPR)=$\frac{FP}{FP+TN}$ and True Positive Rate(TPR)=$\frac{TP}{TP+FN}$. A good classifier should have a very high value(nearly 1) of TPR and low value of FPR(nearly 0).

4 Implementation

In this section, we briefly mention step by step how we are going to proceed to conduct our experiments. As more or less for all four methods steps are almost same, I am trying to make a generalized progress of our experiment so that any machine learning and deep learning technique can be applicable to that.

1. Load the Train and Test dataset and have a quick view of first 10-15 rows so that it would be understandable which columns are quantitative and which are qualitative. Also, which column is the decision/class variable.
2. Next, we do preliminary data preprocessing as described in subsection 2.2 on 2. Note that, for KNN and ANN we don't need feature selection step.
3. Thirdly, we implement the machine learning method to train the data and try to observe with different optimized/validation function both mathematically and if possible by using pictorial representation that which parameter/s(/estimation method) is/are suitable for the data.
   For example, to implement ANN successfully we need to choose activate function and number of neurons for inside layers. Also we need to choose another activation function for final or output layer. From the course, we

have known that for binary classification we can consider 'sigmoid' or 'tan-hyperbolic' function for output layer and for input layer we consider 'Relu'.

4. Finally, when final parameter has been decided we now can make a comparison table for all the four techniques that we have mentioned in the previous section.

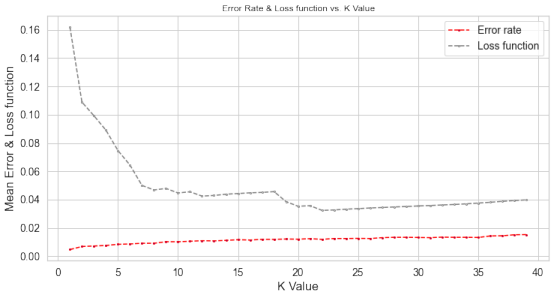
5 Experimental results

In this section, we briefly mentioned our obtained result regarding the implementation of four methods that we mentioned in our previous sections. We divide our experimental outcomes into two ways. First, we show in terms of suitable plots how to the optimal parameters corresponding to KNN and ANN methods can be chosen in terms of least binary cross entropy and highest accuracy value. Secondly, a comparison tabulated representation is provided with the validation measures, namely True Positive Rate(TPR), False Positive Rate(FPR), Binary-cross entropy and Accuracy of the classification.

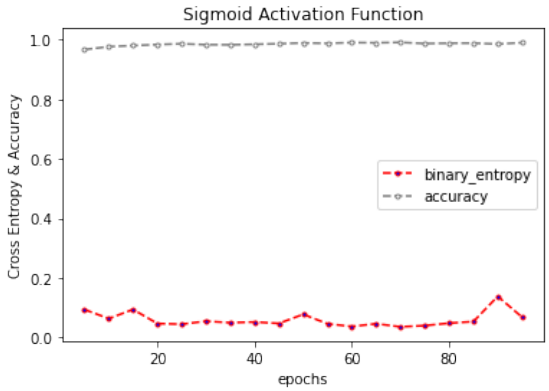
Table 1: Comparison among five network intrusion detectors

Methods Implemented	Validation measure			
	TPR	FPR	Cross-entropy loss	Accuracy
Logistic regression	0.969289	0.057617	0.11882	0.95682
SVM	0.852821	0.018661	0.575747	0.846885
KNN	0.991346	0.015549	0.032381	0.988092
ANN('sigmoid')	0.985132	0.0100645	0.0566	0.987371
ANN('tanh')	0.994143	0.042323	0.2836	0.977147

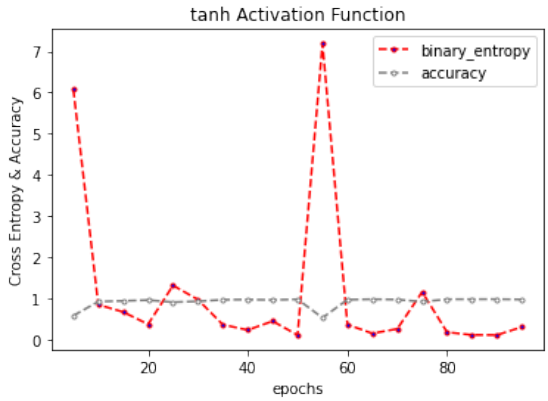
From the figure 4a it has been decided that optimal choice of K-value for KNN method is 22. On other hand, figures 4b 4c suggest that the optimal choice of number of epochs corresponding to ANN('sigmoid') is 60, while the same is for ANN('tanh') is 85.
From the table 1, it is quiet clear that in terms of FPR, Binary cross-entropy loss and accuracy ANN('sigmoid') and KNN perform better than the other machine learning techniques.



(a) KNN method



(b) ANN with 'sigmoid' activation function



(c) ANN with 'tanh' activation function

Fig. 4: Figure showing how the optimal parameter of K-nearest neighbor and Multi-layer artificial neural network has been chosen with respect to loss function and accuracy measure

6 Conclusions

In this project we make an attempt to observe how different machine learning and deep learning-based techniques perform as intrusion detectors. But, there is always a room of improvement in such experimental research findings. For example, we may consider a large variety of number of neurons for input layers and also can increase the size of train dataset to get a better model which may provide better accuracy and False positive rate.

In future, this research can also be extended by adding more machine learning techniques,viz. Decision Trees, Hidden Markov Model, Genetic technologies etc. and more variety of intrusion datasets so that the robustness of our model can be challenged and scope of improvement will rise in terms of producing an ideal AIDS.

References

Arul et al., 2013. Arul, A., Subburathinam, K., and Sivakumari, S. (2013). Classification techniques for intrusion detection an overview. *International Journal of Computer Applications*, 76:33–40.

Bishop, 2006. Bishop, C. M. (2006). Pattern recognition and machine learning. page 209.

Chen et al., 2005. Chen, W.-H., Hsu, S.-H., and Shen, H.-P. (2005). Application of svm and ann for intrusion detection. *Computers and Operations Research*, 32(10):2617–2634. Applications of Neural Networks.

Co, 2002. Co, J. P. A. (2002). *Computer Security Threat Monitoring and Surveillance*.

Defazio et al., 2014. Defazio, A., Bach, F. R., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *CoRR*, abs/1407.0202.

Khraisat et al., 2019. Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2.

Liao et al., 2013. Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., and Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24.

McLeod, 2011. McLeod, I. (2011). k-nearest neighbor (knn) classifier.

Meza et al., . Meza, J., Campbell, S., and Bailey, D. Mathematical and statistical opportunities in cyber security.

Schmidt et al., 2013. Schmidt, M., Roux, N. L., and Bach, F. (2013). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*.