Research and Application of Key Technologies for Network Security Situational Awareness Based on Machine Learning

Hongbin Gu, Xiaolong Li, Xi Yang, and Kongpeng Wei(✉)

Panjin Vocational and Technical College, No. 999 Zhengbang Road, Liaodong Bay New District, Panjin 124000, Liaoning, China
kpwei@pjvtc.edu.cn

Abstract. With the emergence of a large number of network attacks due to the booming network, it is no longer possible to meet the higher requirements of network security if we only rely on simple network security protection technologies such as access control, firewall, intrusion detection, etc. In this situation, the research of network security situational awareness is increasingly applied to network security protection. Network security situational awareness is an environment-based, dynamic, and holistic insight into security risks. Based on the conceptual model of situational awareness, this paper introduces the key technologies of network security situational awareness, and compares the evaluation and prediction effects of different machine learning algorithms applied to network security situational awareness based on the massive network security situational data from multiple heterogeneous sources.

Keywords: Machine learning · Network security · Situational awareness

1 Introduction

Situational awareness technology is a powerful monitoring and assurance technology for network security. Facing the problems that traditional network security technology cannot better detect the network state and explore its change pattern, this paper combines the advantages of machine learning in big data analysis and prediction, and uses the publicly available CIC-IDS-2017 dataset and KDDCUP99 dataset to compare different machine learning algorithms in terms of network security situational awareness assessment and prediction effects and data training time consumption.

1.1 Concepts Related to Network Security Situational Awareness

Situation Awareness (SA) is defined for the first time by Endsley as “the awareness and understanding of environmental factors and the prediction of future trends in a certain spatial and temporal context”.

The literature [1] discusses the concept of network security situational awareness, which is "the acquisition, understanding, display, and prediction of future trends of

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 1566–1572, 2023.
https://doi.org/10.2991/978-94-6463-198-2_163

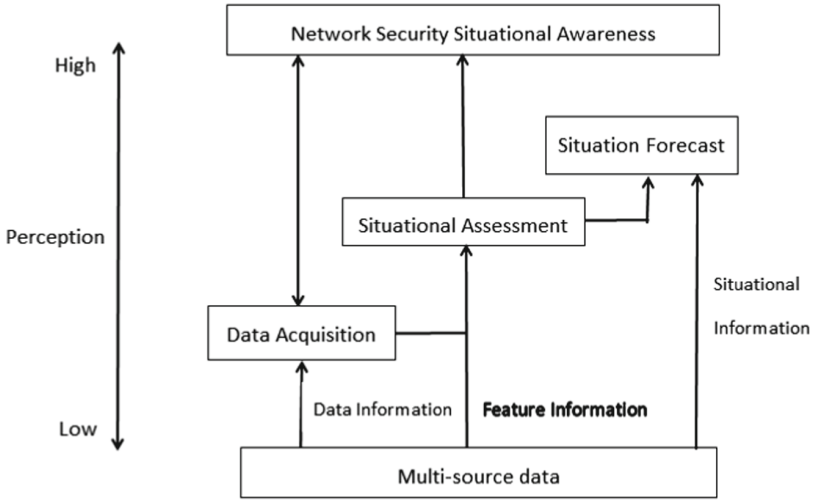

Fig. 1. Situational Awareness Model

security elements that can cause changes in network posture in a large-scale network environment.

Foreign research work on network security situational awareness was conducted earlier and relatively systematic, as early as 1988, when Endsley defined network security situational awareness as a 3-step process, i.e., "acquisition of network elements, situational understanding, and prediction of the future in a specific spatial and temporal environment of the network", as shown in Fig. 1.

1.2 Key Technologies for Network Security Situational Awareness

The literature [2] has a large discussion on algorithms for network security posture assessment, and he classifies the algorithms for network security posture assessment into the following categories: fusion methods based on logical relationships, fusion methods based on mathematical models, fusion methods based on probabilistic statistics, and fusion methods based on rule-based reasoning. For network security posture prediction, methods such as neural networks, time series prediction methods, and support vector machines are generally used.

The literature [3] classifies algorithms for network security posture assessment into the following three categories: knowledge inference methods, statistical methods, and gray scale theory methods.

The literature [4] classifies the key techniques for network security situational awareness into situational awareness methods based on hierarchical analysis, machine learning, immune systems, and game theory.

From the above three review articles, it can be seen that in the early days of network security situational awareness research, methods such as neural networks, time series prediction methods, and support vector machines, which belong to machine learning, were only used in network security situational prediction.

The literature [5–10] uses different machine learning techniques such as support vector machines, neural networks, GRU, LightGBM, XGBoost, LSTM, and Bi-LSTM on different datasets for network security situational awareness.

2 Research on Network Security Situational Awareness Technology Based on Machine Learning

2.1 Machine Learning

The purpose of machine learning is to train a computer to perform certain operations (by running machine learning algorithms) from data to "learn" from previous situations (training data sets) and to enable the computer to make improved decisions in the future. Machine learning is widely used in data mining, where large amounts of data are used to find unknown or hidden patterns.

Machine learning can be divided into two main categories: supervised learning and unsupervised learning. In supervised learning datasets, all training and test data are given a value, i.e., a label, which can be either a numerical value or a character value. Machine learning algorithms are used to learn and predict the labels of unknown data. In unsupervised learning, the data is not labeled and machine learning algorithms are needed to find the pattern of the data and predict the unknown data.

2.2 Network Security Situational Awareness Dataset

Research on network security situational awareness using machine learning requires datasets with large amounts of data. The datasets can use private datasets from companies or the public sector, or datasets that are already publicly available. The advantage of using publicly available datasets is that many publicly published research results use publicly available datasets, which allows new researchers to conduct further research based on the same foundation as previous research results, and facilitates peer-to-peer communication. Publicly available network security datasets include: CIC-IDS-2017 dataset, HoneyNet-data dataset, KDD CUP99 dataset, etc.

The dataset used in this paper is the CIC-IDS-2017 dataset. A collaborative project between the Communications Security Establishment (CSE) and the Canadian Institute for Network security (CIC), which evaluated 11 datasets available since 1998, showed that most of them (e.g., the classic KDDCUP99, NSLKDD, etc.) are outdated and unreliable. Some of these datasets lack traffic diversity and capacity, some do not cover a variety of known attacks, while others anonymize packet payload data, which does not reflect current trends. Some also lack feature sets and metadata.

The CIC-IDS-2017 dataset contains benign and up-to-date common attacks, similar to real-world data (PCAPs).

It was collected as of 5 p.m. on Friday, July 7, 2017, for a total of 5 days. Monday is a normal day and includes only normal traffic. The implemented attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attacks, Infiltration, Botnets and DDoS. They were executed on Tuesday, Wednesday, Thursday and Friday morning and afternoon respectively.

To compare the result of different datasets with machine learning algorithm, we use KDDCUP99 also. It is a dataset collected in 1998.

2.3 Data Pre-processing

The pre-processing of data includes normalization of data and normalization of data, label coding.

Normalization of data is to scale the value of a column of features in the training set and test set to between 0 and 1. The method is shown below.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Normalization of the data is to reduce the values of a column of features in the training and test sets to a mean of 0 and a variance of 1. The method is shown below.

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

In scikit-learn, the most common third-party module for machine learning, the normalization and normalization of both training and test data can be implemented in a single line of code.

$$\text{X\_train} = (\text{X\_train} - \text{X\_train.mean}())/\text{X\_train.std}()$$
$$\text{X\_test} = (\text{X\_test} - \text{X\_test.mean}())/\text{X\_test.std}()$$

Since machine learning algorithms process data mostly in the form of matrix representation of numerical values for the calculation of string features need to be converted to numbers. There are two mainstream methods for converting strings to numerical values, one is tag encoding and the other is unique hot encoding. Since the strings to be converted in the CIC-IDS-2017 dataset are class identifiers, we use tag encoding, which is implemented in scikit-learn as follows.

$$\text{labelencoder} = \text{LabelEncoder}()$$
$$\text{y\_train} = \text{labelencoder.fit\_transform(df\_train.iloc[:, [80]])}$$

2.4 Comparison of the Effectiveness and Training Time of Different Machine Learning Methods

Since different regression and classification algorithms are encapsulated in scikit-learn, the same training and test sets are trained and predicted with different algorithms, and the comparison results are obtained as shown in Table 1.

As can be seen from the table, these algorithms differ significantly in mean absolute error and R2. Among them, neural network and support vector machine perform well on R2, indicating better model fitting, but the training time of neural network is two orders of magnitude more than the other algorithms, and the training time of support vector machine is three orders of magnitude more than the other algorithms. The three algorithms, stochastic gradient descent, linear regression, and ridge regression, performed moderately well on R2, and the other algorithms performed poorly on R2. For the deployed model, of course, the neural network with the best performance is

Table 1. Comparison of the effectiveness and training time of different machine learning methods

	RMSE	R2	Training Time (Second)
Neural Networks	0.0017	0.9930	16.8188
Support vector machines	0.1355	0.9797	307.96
Stochastic gradient descent	0.0283	0.8845	0.2293
Linear regression	0.0298	0.8785	1.8365
Ridge regression	0.1755	0.7144	0.0489
Lasso regression	0.2952	0.5516	0.0965
Elastic network regression	0.4908	−0.000031	0.0495
Decision trees	0.5574	−1.2735	0.4986
Plain Bayesian	0.5693	−1.32202	0.0787

chosen, but in the training process, due to the limitation of computing power, complex models take more time on larger data volumes, and probably stochastic gradient descent is a better choice in case of insufficient computing power, taking into account the model fitting effect and training time. With another dataset - KDDCUP99, the RMSE of different machine learning algorithms are in a narrow range of 0.1355 to 0.1738 and the R2 are in the range of 0.2715 to 0.3158 which show that there are not much difference in different algorithms.

The code for training and testing different algorithms in scikit-learn is similar, taking ridge regression as an example, the code is as follows.

```
# Ridge regression
from sklearn import linear_model
RidgeReg = linear_model.Ridge(alpha=.5)
# Record training start time
start = time.perf_counter()
# Model Training
RidgeReg.fit(X_train, y_train)
# Calculate the time spent on training
elapsed_time = time.perf_counter() - start
print('train_time  : {0}'.format(elapsed_time) + ' [sec]')
# Model Predictions
y_pred = RidgeReg.predict(X_test)
# Effectiveness evaluation
evaluation(y_test,y_pred,index_name=['RidgeReg'])
```

The algorithm effects and training times quoted in the above code are represented in the following functional form.

```
def evaluation(y_true,y_pred,index_name=['OLS']):
    df=pd.DataFrame(index=[index_name],columns=['RMSE'])
    df["RMSE"]=metrics.mean_squared_error(y_true,y_pred)
    return df
```

The above codes also include the method of prediction such as ridge regression:

```
y_pred = RidgeReg.predict(X_test)
```

3 Conclusion

After several years of rapid development, machine learning technology has become increasingly mature, and in the application of machine learning algorithms to network security situational awareness assessment and prediction, the models in existing third-party modules can be used to conveniently train and test data and predictions. By comparing the evaluation effects of different models, it is found that more complex algorithms such as neural networks have a better model fit, and if the amount of data is large and the computing power is limited, moderately complex algorithms such as stochastic gradient descent algorithms can strike a balance between model fit and training time.

References

1. Abasi. A Network Security Situational Awareness Model Based on Information Fusion[J]. Journal of Computer Research & Development, 2009, 846–847:1632–1635.Davis, A. R., Bush, C., Harvey, J. C. and Foley, M. F., "Fresnel lenses in rear projection displays," SID Int. Symp. Digest Tech. Papers 32(1), 934–937 (2001).
2. Ying Z , Qiang Z , Gong Z . Cyberspace Situational Awareness Model based on Spatial Traffic Clustering.
3. Li W . Analysis framework of network security situational awareness and comparison of implementation methods[J]. Eurasip Journal on Wireless Communications and Networking, 2019, 2019(1).
4. Ali J R , Dadashtabar A K , Samsami K F . PROJECTION OF MULI STAGE CYBER ATTACK BASED ON BELIEF MODEL AND FUZZY INFERENCE. 2015.FamilyName, GivenName Initial., "Title," Source, pg# (year).
5. The Model of Network Security Situational Awareness[C]// 2011.
6. Ying L , Li B , Wang H . Dynamic awareness of network security situation based on stochastic game theory[C]// 2010.
7. Yue G , Zhang S . A Network Security Situation Awareness Method Based on Multi- source Information Fusion[C]// 2nd International Forum on Management, Education and Information Technology Application (IFMEITA 2017). 2018.
8. Bian N , Wang X , Li M . Network security situational assessment model based on improved AHP_FCE[C]// International Conference on Advanced Computational Intelligence. IEEE, 2013.

9. Zhang D , Qian K , Wang W , et al. Network Security Situation Awareness Technology Based on Multi-source Heterogeneous Data[C]// CIAT 2020: 2020 International Conference on Cyberspace Innovation of Advanced Technologies. 2020.
10. Damarla T . Hidden Markov model as a framework for situational awareness[C]// International Conference on Information Fusion. IEEE, 2008.