

**International Journal of
Engineering Research and Science & Technology**



ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

CYBER THREAT DETECTION IN CLOUD BASED ON ARTIFICIAL NEURAL NETWORK USING EVENT PROFILES

Ms.Janke.Kalyani¹, Dr. PM Yohan², Dr. Santosh Kumar Byraboina³

¹Assistant professor, Dept of Computer Science, University PG College, Osmania University, Secunderabad.

²Professor and Principal, CSI Wesley Institute of Technology and Sciences, Secunderabad.

³Associate Professor, Department of MCA, Wesley PG College, Secunderabad.

ABSTRACT

Developing an automated method for detecting cyberthreats is one of the main issues facing cyber security. In this paper, we describe an artificial neural network-based cyberthreat detection method. The suggested solution improves cyber-threat identification by converting a large number of gathered security events into unique event profiles and utilizing a deep learning-based detection algorithm. In this study, we created an AI-SIEM system using a variety of artificial neural network techniques, such as CNN, LSTM, and FCNN, together with event profiling for data preparation. The system's primary objective is to distinguish between true positive and false positive alarms, enabling security analysts to act quickly. to online dangers. The authors of this work conducted all of the experiments using two real-world datasets and two benchmark datasets, NSLKDD and CICIDS2017. We ran trials utilizing the five traditional machine-learning techniques (SVM, k-NN, RF, NB, and DT) to assess the performance comparison with current methodologies. As a result, the study's experimental findings confirm that our suggested approaches can be used as learning-based models for network intrusion detection and demonstrate that, despite being used in real-world scenarios, their performance surpasses that of traditional machine learning techniques.

1. INTRODUCTION

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations gives incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear-based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults. Right now, learning the bolster support vector machine (SVM)

calculations were utilized to recognize port sweep endeavors dependent on the new CICIDS2017 dataset with 97.80%, 69.79% precision rates were accomplished individually. Rather than SVM we can introduce some other algorithms like random forest, CNN, ANN where these algorithms can acquire accuracies like SVM – 93.29, CNN – 63.52, Random Forest – 99.93, ANN – 99.11.

1.1 MOTIVATION

The use of new innovations gives incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear-based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.

1.2 OBJECTIVES

Objective of this project is to detect cyber-attacks by using machine learning algorithms like

1.2.1 CNN:

A Convolutional Neural Network (CNN) is a type of deep learning neural network that is well-suited for image and video analysis. Random forest: It is the tech industry's definitive destination for sharing compelling, first-person accounts of problem-solving on the road to innovation.

1.2.2 ANN:

ANN algorithm accepts only numeric and structured data. Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN) are used to accept unstructured and non-numeric data forms such as Image, Text, and Speech

1.3 OUTCOMES:

These predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber-attacks happened or not.

1.4 APPLICATIONS:

This strategy used in Detection of Cyber Attack in Network using Machine Learning Techniques

2. LITERATURE SURVEY

Literature Survey Intrusion detection is very important for network situation awareness. While a few methods have been proposed to detect network intrusion, they cannot directly and effectively utilize semi-quantitative information consisting of expert knowledge and quantitative data.

Hence, this paper proposes a new detection model based on a directed acyclic graph (DAG) and a belief rule base (BRB). In the proposed model, called DAG-BRB, the DAG is employed to construct a multi-layered BRB model that can avoid explosion of combinations of rule number because of a large number of types of intrusion. To obtain the optimal parameters of the DAG-BRB model, an improved constraint covariance matrix adaption evolution strategy

(CMA-ES) is developed that can effectively solve the constraint problem in the BRB. A case study was used to test the efficiency of the proposed DAG-BRB.

The results showed that compared with other detection models, the DAG-BRB model has a higher detection rate and can be used in real network.

Nowadays, IT organizations generate colossal amounts of data. Handling these chunks of data itself is critical in the IT world. Hence centralizing the log management system improves security thereby enhances data protection in an organization. Such enterprises require a high profiling tool that helps in managing the information and events data to improve the level of security.

Security Information and Event Management (SIEM) :

It is a procedure for security analysis that prominence an overview of security in an organization. SIEM tools collect, analyze, normalize and correlates all files and analyze data coming from the various device and give a centralized view of logs. This paper articulates an abstraction of SIEM tools and event correlation engines, furnishing a description of their technical comparative study, focusing on most popular SIEM tools and open source rule-based correlation engines and profiles them.

Distributed computing has become an effective approach to enhance capabilities of an institution or organization and minimize requirements for additional resource. In this regard, the distributed computing helps in broadening institutes IT capabilities. One needs to note that distributed computing is now integral part of most expanding IT business sector. It is considered novel and efficient means for expanding business. As more organizations and individuals start to use the cloud

Literature Survey Intrusion detection is very important for network situation awareness. While a few methods have been proposed to detect network intrusion, they cannot directly and effectively utilize semi-quantitative information consisting of expert knowledge and quantitative data. Hence, this paper proposes a new detection model based on a directed acyclic graph (DAG) and a belief rule base (BRB). In the proposed model, called DAG-BRB, the DAG is employed to construct a multi-layered BRB model that can avoid explosion of combinations of rule number because of a large number of types of intrusion.

To obtain the optimal parameters of the DAG-BRB model, an improved constraint covariance matrix adaption evolution strategy (CMA-ES) is developed that can effectively solve the constraint problem in the BRB. A case study was used to test the efficiency of the proposed DAG-BRB. The results showed that compared with other detection models, the DAG-BRB model has a higher detection rate and can be used in real network.

Nowadays, IT organizations generate colossal amounts of data. Handling these chunks of data itself is critical in the IT world. Hence centralizing the log management system improves security thereby enhances data protection in an organization. Such enterprises require a high profiling tool that helps in managing the information and events data to improve the level of security. Security Information and Event Management (SIEM) is a procedure for security analysis that prominence an overview of security in an organization. SIEM tools collect, analyze, normalize and correlates all files and analyze data coming from the various device and give a centralized view of logs.

This paper articulates an abstraction of SIEM tools and event correlation engines, furnishing a description of their technical comparative study, focusing on most popular SIEM tools and open

source rule-based correlation engines and profiles them. Distributed computing has become an effective approach to enhance capabilities of an institution or organization and minimize requirements for additional resource. In this regard, the distributed computing helps in broadening institutes IT capabilities. One needs to note that distributed computing is now integral part of most expanding IT business sector. It is considered novel and efficient means for expanding business. As more organizations and individuals start to use the cloud to store their data and applications, significant concerns have developed to protect sensitive data from external and internal attacks over internet.

Due to security concern many clients hesitate in relocating their sensitive data on the clouds, despite significant interest in cloud-based computing. Security is a significant issue, since data much of an organizations data provides a tempting target for hackers and those concerns will continue to diminish the development of distributed computing if not addressed. Therefore, this study presents a new test and insight into a honeypot.

It is a device that can be classified into two types:

1. Handling
2. research honeypots.

Handling honeypots are used to mitigate real life dangers:

A research honeypot is utilized as an exploration instrument to study and distinguish the dangers on the internet. Therefore, the primary aim of this research project is to do an intensive network security analysis through a virtualized honeypot for cloud servers to tempt an attacker and provide a new means of monitoring.

3. EXISTING SYSTEM

1.1 EXISTING APPROACH:

Blameless Bayes and Principal Component Analysis (PCA) were been used with the KDD99 dataset by Almansob and Lomte. Similarly, PCA, SVM, and KDD99 were used Chithik and Rabbani for IDS. In Aljawarneh et Al's. Consequently, KDD99 is old and doesn't give any data about cutting edge new assault types, example, multi day misuses and so forth.

1.1.1 Drawbacks

- Strict Regulations
- Difficult to work with for non-technical users
- Restrictive to resources
- Constantly needs Patching
- Constantly being attacked

4. PROPOSED SYSTEM

4.1 Proposed System

Important steps of the algorithm are given in below.

- Normalization of every dataset.
- Convert that dataset into the testing and training.

- Form IDS models with the help of using AI models RF, ANN, CNN and SVM algorithms.
- Evaluate every model's performance

4.2 ALGORITHMS

CNN:

A Convolutional Neural Network (CNN) is a type of deep learning neural network that is well-suited for image and video analysis.

Random forest:

It is the tech industry's definitive destination for sharing compelling, first-person accounts of problem-solving on the road to innovation.

ANN:

ANN algorithm accepts only numeric and structured data. Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN) are used to accept unstructured and non-numeric data forms such as Image, Text, and Speech

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of factors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

4 RESULTS

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

import itertools
import seaborn as sns
import pandas_profiling
import statsmodels.formula.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from patsy import dmatrices

/usr/local/lib/python3.6/dist-packages/statsmodels/tools/testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
import pandas.util.testing as tm

from sklearn import datasets
from sklearn.feature_selection import RFE
import sklearn.metrics as metrics
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2, f_classif, mutual_info_classif

train=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Train.txt',sep=',')
test=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Test.txt',sep=',')
    
```

Fig 4.1:Input Data

```

n [6]: columns=["duration","protocol_type","service","flag","src_bytes","dst_bytes","land",
"wrong_fragment","urgent","hot","num_failed_logins","logged_in",
"num_compromised","root_shell","su_attempted","num_root","num_file_creations",
"num_shells","num_access_files","num_outbound_cmds","is_host_login",
"is_guest_login","count","srv_count","serror_rate","srv_error_rate",
"error_rate","srv_error_rate","same_srv_rate","diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_s
"dst_host_diff_srv_rate","dst_host_same_src_port_rate",
"dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_error_rate",
"dst_host_rerror_rate","dst_host_srv_rerror_rate","attack", "last_flag"]

n [7]: train.columns=columns
test.columns=columns

n [8]: train.head()
ut [8]:
    duration  protocol_type  service  flag  src_bytes  dst_bytes  land  wrong_fragment  urgent  hot  num_failed_logins  logged_in  num_compromised  root_
0          0             udp    other   SF         146         0         0             0         0         0             0           0             0
1          0             tcp    private SO           0           0         0             0         0         0             0           0             0
2          0             tcp     http   SF         232        8153         0             0         0         0             0           1             0
3          0             tcp     http   SF          199         420         0             0         0         0             0           1             0
4          0             tcp    private REJ           0           0         0             0         0         0             0           0             0

n [9]: test.head()
    
```

Fig 4.2: Data Processing

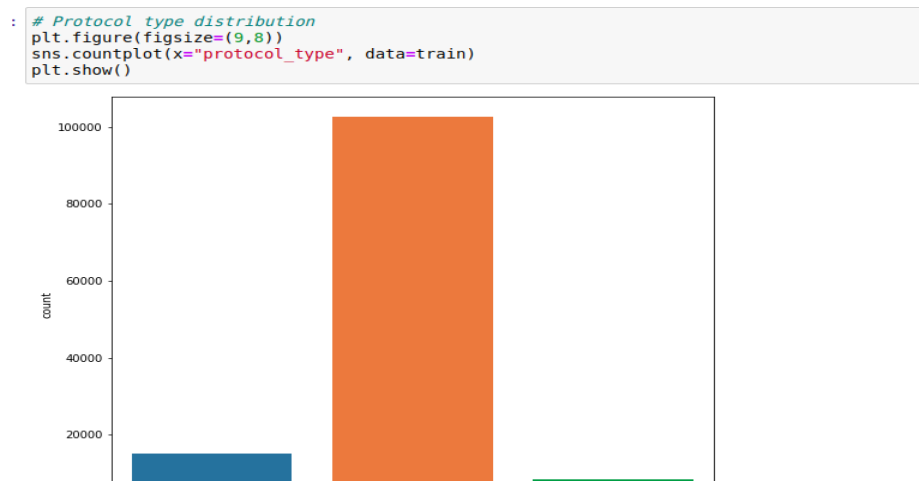


Fig 4.3: DATA EDA

Model Building

```
train_X=train_new[cols]
train_y=train_new['attack_class']
test_X=test_new[cols]
test_y=test_new['attack_class']
```

Fig 4.4: Model Building

Logistic Regression

```
# Building Models
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state=0,solver='lbfgs',multi_class='multinomial')
logreg.fit( train_X, train_y)
logreg.predict(train_X) #by default, it use cut-off as 0.5

list( zip( cols, logreg.coef_[0] ) )

logreg.intercept_

logreg.score(train_X,train_y)
```

Fig 4.5: Logistic Regression

Decision Trees

```
train_X.shape

param_grid = {'max_depth': np.arange(2, 12),
              'max_features': np.arange(10,15)}

train_y.shape

from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier, export_graphviz, export
tree = GridSearchCV(DecisionTreeClassifier(), param_grid, cv = 10,verbose=1,n_jobs=-1)
tree.fit( train_X, train_y)

tree.best_score_

tree.best_estimator_
tree.best_params_

train_pred = tree.predict(train_X)

print(metrics.classification_report(train_y, train_pred))

test_pred = tree.predict(test_X)
```

Fig 4.6: Decision Trees

Random Forest

```
: from sklearn.ensemble import RandomForestClassifier
pargrid_rf = {'n_estimators': [50,60,70,80,90,100],
             'max_features': [2,3,4,5,6,7]}

: from sklearn.model_selection import GridSearchCV
gscv_rf = GridSearchCV(estimator=RandomForestClassifier(),
                      param_grid=pargrid_rf,
                      cv=10,
                      verbose=True, n_jobs=-1)

gscv_results = gscv_rf.fit(train_X, train_y)

: gscv_results.best_params_

: gscv_rf.best_score_

: radm_clf = RandomForestClassifier(oob_score=True,n_estimators=80, max_features=5, n_jobs=-1)
radm_clf.fit( train_X, train_y )

: radm_test_pred = pd.DataFrame( { 'actual': test_y,
                                  'predicted': radm_clf.predict( test_X ) } )
```

Fig 4.7: Random forest


```
Support Vector Machine (SVM)

: from sklearn.svm import LinearSVC
svm_clf = LinearSVC(random_state=0, tol=1e-5)
svm_clf.fit(train_X,train_y)

: print(svm_clf.coef_)
print(svm_clf.intercept_)
print(svm_clf.predict(train_X))

: from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline

model = SVC(kernel='rbf', class_weight='balanced',gamma='scale')

: model.fit(train_X,train_y)

: from sklearn.model_selection import GridSearchCV
param_grid = {'C': [1, 10],
              'gamma': [0.0001, 0.001]}
grid = GridSearchCV(model, param_grid)
grid.fit(train_X,train_y)

: print(grid.best_params_)
```

Fig 4.8: Support Vector machine (SVM)

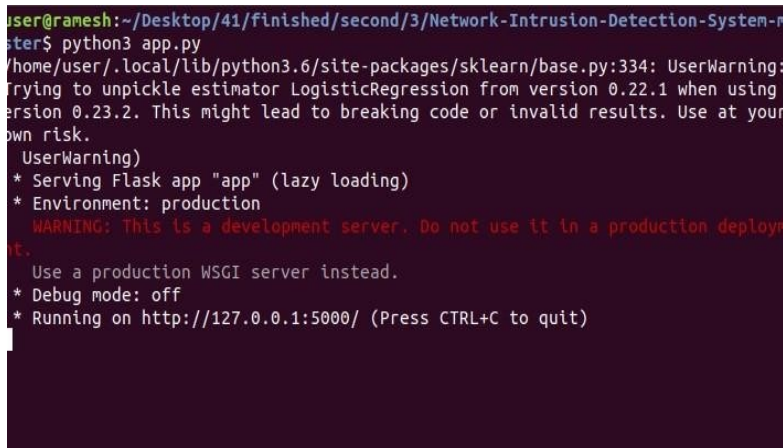


Fig 4.9: LocalHost in cmd python app.py

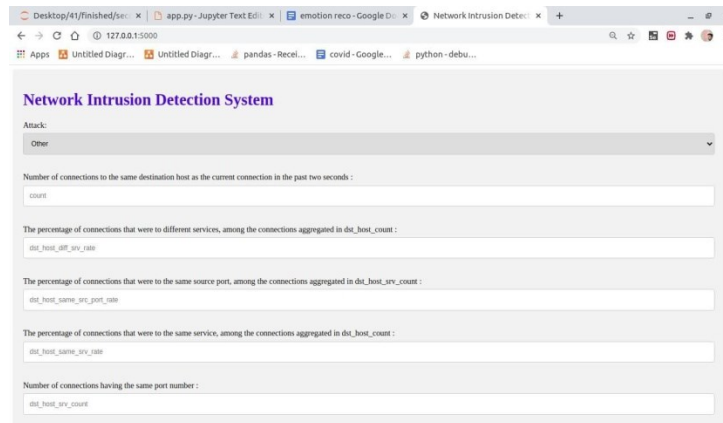


Fig 4.10: Cloud Network Intrusion Detection System

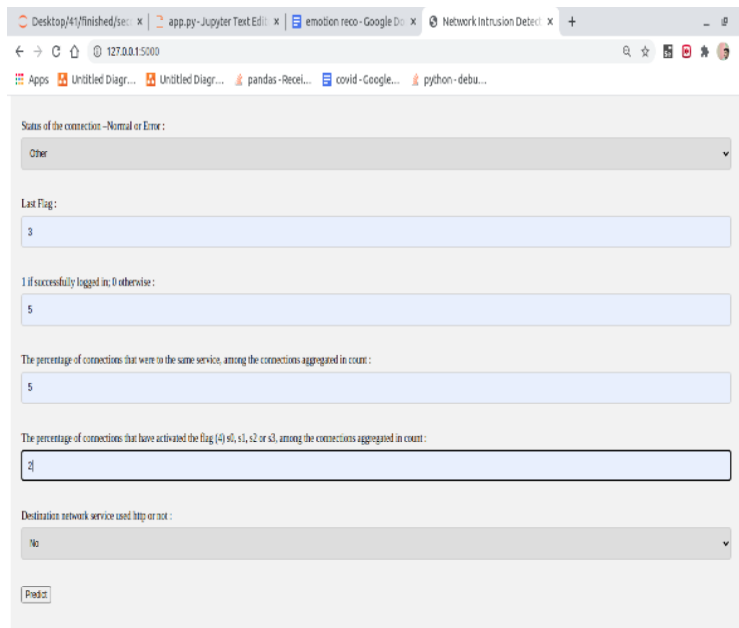


Fig 4.11: Entering the input data

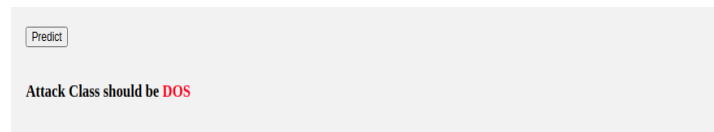


Fig 4.12: Predict attack

5. CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

As of right now, estimates for deep learning computations, ANN, CNN, Random Forest, and support vector machines rely on the current CICIDS2017 dataset, which was introduced in a pretty light manner. The profound learning computation produced results that were fundamentally better than those of SVM, ANN, RF, and CNN, according to the results. Depending on this dataset, we will use port sweep attempts along with additional attack types utilizing AI and deep learning computations, Apache Hadoop, and Spark technologies. All of these calculations aid in the detection of network cyberattacks. It occurs in such a way that, if we go back a long way, there may have been a great deal of attacks, so when these attacks are identified, certain datasets will contain the characteristics at which these attacks are occurring. Therefore, we will be able to forecast whether or not a cyberattack will occur by using these datasets. These forecasts can be produced using four different algorithms: SVM, ANN, RF, and CNN. In order to forecast the best outcomes for determining whether or not cyberattacks have occurred, this study assists in identifying which algorithm has the highest accuracy rates. With possible advantages in terms of detection accuracy, real-time monitoring, adaptive learning, and scalability, the future potential of cyber threat detection utilizing ANNs based on event profiles seems encouraging. To effectively install and manage these cutting-edge systems, firms must take into account the obstacles and make the appropriate investments in resources.

REFERENCES

- [1] K. Graves, *Ceh: Official certified ethical hacker review guide: Exam 312-50*. John Wiley & Sons, 2007.
- [2] R. Christopher, "Port scanning techniques and the defense against them," SANS Institute, 2001.
- [3] M. Baykara, R. Das, and I. Karadogan, "Bilgi güvenli ğisistemlerindekullanılanarac, larinincelenmesi," in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.
- [4] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *Journal of Computer Security*, vol. 10, no. 1-2, pp. 105–136, 2002.
- [5] S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, "Surveillance detection in high bandwidth environments," in DARPA Information Survivability Conference and Exposition, 2003. Proceedings, vol. 1. IEEE, 2003, pp. 130–138.
- [6] K. Ibrahim and M. Ouaddane, "Management of intrusion detection systems based-kdd99: Analysis with lda and pca," in *Wireless Networks and Mobile Communications (WINCOM), 2017 International Conference on*. IEEE, 2017, pp. 1–6.
- [7] N. Moustafa and J. Slay, "The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems," in *Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on*. IEEE, 2015, pp. 25–31.
- [8] L. Sun, T. Anthony, H. Z. Xia, J. Chen, X. Huang, and Y. Zhang, "Detection and classification of malicious patterns in network traffic using Binford's law," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*. IEEE, 2017, pp. 864–872.
- [9] S. M. Almansob and S. S. Lomte, "Addressing challenges for intrusion detection system using naive bayes and pca algorithm," in *Convergence in Technology (I2CT), 2017 2nd International Conference for*. IEEE, 2017, pp. 565–568.
- [10] M. C. Raja and M. M. A. Rabbani, "Combined analysis of support vector machine and principle component analysis for ids," in *IEEE International Conference on Communication and Electronics Systems*, 2016, pp. 1–5.
- [11] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.
- [12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.
- [13] D. Aksu, S. Ustebay, M. A. Aydin, and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm," in *International Symposium on Computer and Information Sciences*. Springer, 2018, pp. 141–149.
- [14] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark," *IEEE Access*, 2018.
- [15] P. A. A. Resende and A. C. Drummond, "Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling," *Security and Privacy*, vol. 1, no. 4, p. e36, 2018.
- [16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] R. Shouval, O. Bondi, H. Mishan, A. Shimoni, R. Unger, and A. Nagler, "Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in sct," Bone marrow transplantation, vol. 49, no. 3, p. 332, 2014.