International Journal of HRM and Organizational Behavior

www.ijhrmob.com

editor@ijhrmob.com

International Journal of
HRM and Organizational Behavior

ISSN 2454 - 5015

DEEP GENERATIVE LEARNING MODELS FOR CLOUD INTRUSION DETECTION SYSTEMS

G.SRUJAN KUMAR, Assistant Professor, Dept of MCA, Chirala Engineering College, Chirala,
srujan9032@gmail.com

VANKAYALAPATI ANUSHA, PG student - MCA, Dept of MCA, Chirala Engineering College, chirala,
Anushavankayalapati163@gmail.com

Abstract: The project centers around the development and utilization of deep generative learning models specifically tailored for Cloud Intrusion Detection Systems (IDS), aiming to address the challenge of effectively detecting unknown attacks within the cloud environment. The proposed solution involves leveraging two specific deep generative models: the conditional denoising adversarial autoencoder (CDAAE) and the hybrid model CDAAE-KNN, each serving a unique purpose in generating malicious samples. The conditional denoising adversarial autoencoder (CDAAE) is employed to generate targeted types of malicious samples, aiding in the augmentation of the dataset for training the cloud IDS. The hybrid model CDAAE-KNN is utilized to generate malicious borderline samples, which are crucial in refining the IDS's accuracy by focusing on samples that reside near the decision boundary. The malicious samples generated by CDAAE and CDAAE-KNN are integrated with the original dataset, creating augmented datasets with a richer diversity of samples that encompass both specific malicious types and borderline cases. Three machine learning algorithms are trained on the augmented datasets to evaluate their effectiveness and performance in detecting intrusions within the cloud environment. This step aims to comprehensively analyze the impact of the generated samples on the IDS's accuracy and robustness.The project extends its capabilities with the integration of a Stacking Classifier, combining Extratree Classifier and LinearSVC with Logistic Regression, to enhance the accuracy and robustness of intrusion detection. This ensemble approach demonstrates superior performance in identifying potential security threats within cloud environments.

Index terms *-Cloud systems, conditional denoising adversarial autoencoder (CDAAE), deep learning, generative models, intrusion detection (ID).*

1. INTRODUCTION

Due to the huge economic benefit, cloud computing mar- ket has been experiencing an unprecedented development over the last 5 years. Today, the global cloud computing market is worth $180B in revenue with the annual market growth of 24% [1]. Nevertheless, the wide adoption of cloud computing also resulted in the cloud systems being very vulnerable to many types of cyber attacks. The security of the cloud environment is therefore an increasing concern of both service providers and end users [2]. Among several approaches to protecting the cloud systems, intrusion detection systems (IDSs) play a crucial role in early detecting and preventing security attacks [3]–[6].

Recently, IDS researchers have paid more attention to developing techniques for handling distributed denial of service (DDoS) attacks [7]. This is due to the wide spread of DDoS attacks and serious impact they cause to the availability and the reputation of cloud providers. Among several DDoS attacks, low-rate attacks and application layer level or gateway level attacks are among the most dangerous attacks. These attacks often attempt to conceal themself by mimicking the normal network pattern. For example, DDoS low rate attacks inject low volume legitimate traffic at a very slow rate and these attacks can be conducted using a lesser number of machines.

Since the traffic volume of these attacks is very low and they often appear to be legitimate, the traditional detection methods may fail to detect them [8]. Application layer attacks present another sophisticated version of DDoS attacks in which the attack traffic attempts to be more similar to normal user traffic and hence, pose a serious challenge in how they can be identified. Attackers often use the requests of legitimate users to hide the attacks. As a result, most of the defense techniques at the network layer and application layer fail to detect these attacks [9]. Moreover, these attacks can be executed using multiple protocols at the application layer, both connection oriented and connectionless, making them even more dangerous.

There are two main techniques for identifying the malicious actions on the cloud environment: 1) nonmachine learning approaches and 2) machine learning approaches [10]. Nonmachine learning approaches [10] rely on the characteristics of cloud malicious behaviors to identify the attacks. The advantage of these methods is the short processing time and the ability to accurately detect previously known attacks. The shortcoming is that they highly depend on the knowledge about the signature of attacks and are unable to detect new/unknown types of attacks. Thus, machine learning-based approaches have been developed to overcome the limitation of nonmachine learning approaches.

However, using machine learning to build a trustworthy and robust IDS on the cloud remains practically challenging. One of the reason is the rapid development of various and sophisticated cloud attacks. Another reason is the lack of labeled malicious samples to construct an effective machine learning model. On the cloud environments, majority of collected traffic samples are normal and only a few samples are intrusions. Subsequently, most of the intrusion datasets on the cloud are imbalanced. When being trained on the skewed datasets, the predictive model developed using conventional machine learning algorithms could be biased and hence inaccurate. This is because machine learning algorithms are usually designed to improve the accuracy by reducing the error. Thus, they do not take into account the class distribution/proportion or the balance of classes.

A possible solution to address the above imbalance problem of cloud IDS datasets is collecting more malicious samples. However, on the cloud environments, collecting attack samples is extremely difficult due to the privacy and security concerns of cloud users [11]. Cloud providers tend to avoid divulging data that could compromise the privacy of their clients or privileged information of their systems. Therefore, resampling (oversampling and undersampling) methods are often used to balance the skewed datasets [12]. Nevertheless, these techniques have some intrinsic drawbacks. While undersampling can potentially lose useful information, oversampling is prone to cause overfitting, when exact copies of the minority class are replicated randomly. Moreover, it does not really solve the fundamental "lack of data" problem.

2. LITERATURE SURVEY

Security challenges are still among the biggest obstacles when considering the adoption of cloud services. This triggered a lot of research activities, resulting in a quantity of proposals targeting the various cloud security threats. Alongside with these security issues, the cloud paradigm comes with a new set of unique features, which open the path toward novel security approaches, techniques, and architectures. This paper [2] provides a survey on the achievable security merits by making use of multiple distinct clouds simultaneously. Various distinct architectures are introduced and discussed according to their security and privacy capabilities and prospects.

Internet of Things (IoT) has emerged as a cutting-edge technology that is changing human life. The rapid and widespread applications of IoT, however, make cyberspace more vulnerable, especially to IoT-based attacks in which IoT devices are used to launch attack on cyber-physical systems. Given a massive number of IoT devices (in order of billions), detecting and preventing these IoT-based attacks are critical. However, this task is very challenging due to the limited energy and computing capabilities of IoT devices and the continuous and fast evolution of attackers. [3] Among IoT-based attacks, unknown ones are far more devastating as these attacks could surpass most of the current security systems and it takes time to detect them and "cure" the systems. To effectively detect new/unknown attacks, in this article, we propose a novel representation learning method to better predictively "describe" unknown attacks, facilitating supervised learning-based anomaly detection methods. Specifically, we develop three regularized versions of autoencoders (AEs) to learn a latent representation from the input data [13, 36, 37]. The bottleneck layers of these regularized AEs trained in a supervised manner using normal data and known IoT attacks will then be used as the new input features for classification algorithms. We carry out extensive experiments on nine recent IoT datasets to evaluate the performance of the proposed models. The experimental results demonstrate that the new latent representation can significantly enhance the performance of supervised learning methods in detecting unknown IoT attacks. We also conduct experiments to investigate the characteristics of the proposed models and the influence of hyperparameters on their performance. The running time of these models is about 1.3 ms that is pragmatic for most applications.

To invade a cyber-physical system (CPS) successfully, hackers are prone to simultaneously launching multiple cyber attacks [2] on different sensors in a CPS. However, little attention has been paid to the problem of detecting multiple cyber attacks up to now. Therefore, in this paper [4], we deal with the problem on how to efficiently detect multiple cyber attacks aiming at different sensors in CPSs. To achieve the goal of simultaneously detecting both the number of attacks and the attacked sensors, we formulate this problem via a random finite set (RFS) theory, and then apply an iterative RFS-based Bayesian filter and its approximation to solve the problem. Four numerical experiments with different attacks are provided, and the results have demonstrated the effectiveness of the RFS-based approach for the problem of multiple attacks detection in CPSs.

In this paper [5], a deep neural network (DNN), a type of deep learning model, is explored to develop a flexible and effective IDS to detect and classify unforeseen and unpredictable cyberattacks. The continuous change in network behavior and rapid evolution of attacks makes it necessary to evaluate various datasets which are generated over the years through static and dynamic approaches. This type of study facilitates to identify the best algorithm which can effectively work in detecting future cyberattacks. A comprehensive evaluation of experiments of DNNs and other classical machine learning classifiers are shown on various publicly available benchmark malware datasets. The optimal network parameters and network topologies for DNNs [5] are chosen through the following hyperparameter selection methods with KDDCup 99 dataset [30, 31, 32]. All the experiments of DNNs are run till 1,000 epochs with the learning rate varying in the range [0.01-0.5]. The DNN model which performed well on KDDCup 99 is applied on other datasets, such as NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017, to conduct the benchmark. Our DNN model learns the abstract and high-dimensional feature representation of the IDS data by passing them into many hidden layers. Through a rigorous experimental testing, it is confirmed that DNNs perform well in comparison with the classical machine learning classifiers. Finally, we propose a highly scalable and hybrid DNNs framework called scale-hybrid-IDS-AlertNet which can be used in real-time to effectively monitor the network traffic and host-level events to proactively alert possible cyberattacks.

The rapid increase in network traffic has recently led to the importance of flow-based intrusion detection systems processing a small amount of traffic data. Furthermore, anomaly-based methods, which can identify unknown attacks are also integrated into these systems. In this study, the focus is concentrated on the detection of anomalous network traffic (or intrusions) from flow-based data using unsupervised deep learning methods with semi-supervised learning approach [6]. More specifically, Autoencoder and Variational Autoencoder methods were employed to identify unknown attacks using flow features. In the experiments carried out, the flow-based features extracted out of network traffic data, including typical and different types of attacks, were used. The Receiver Operating Characteristics (ROC) and the area under ROC curve, resulting from these methods were calculated and compared with One-Class Support Vector Machine [7, 9]. The ROC curves were examined in detail to analyze the performance of the methods in various threshold values. The experimental results show that Variational Autoencoder performs, for the most part, better than Autoencoder and One-Class Support Vector Machine.

Wireless sensor networks (WSNs) are commonly adopted by the Industrial Internet-of-Things (IIoT) scenarios due to their easy and fast deployment. However, WSNs are especially vulnerable to security attacks as reflexive packet flooding Denial of Service (DoS) may occur. Admission control and reputation-based strategies are effective for attack detection.

However, mitigation strategies to deal with the effects of these attacks, such as congestion channel transmission is an open issue [7]. Additionally, the resource-constrained nature of nodes, such as the low bandwidth, limited memory, and battery within WSNs, poses a challenge to develop efficient mechanisms in such a scenario. To address this issue, we propose a distributed congestion control by duty-cycle restriction (D-ConCReCT) to detect and mitigate DoS in IIoT. The main goal is to investigate its feasibility in large-scale networks, as well as its ability to reduce both the detection and mitigation times when compared to a previous centralized approach, the so-called congestion control by duty-cycle restriction (ConCReCT). Our results indicate that D-ConCReCT may be applied in the mitigation of DoS attacks in a sensor network scenario with 500 nodes.

3. METHODOLOGY

i) Proposed Work:

The project utilizes CDAAE and CDAAE-KNN [13, 17] deep generative models to create additional malicious samples, enhancing the training dataset for three machine learning algorithms, resulting in an improved cloud-based Intrusion Detection System (IDS) with significantly enhanced accuracy, especially in the detection of challenging Distributed Denial of Service (DDoS) attacks [7, 9].The project extends its capabilities with the integration of a Stacking Classifier, combining Extratree Classifier and LinearSVC with Logistic Regression, to enhance the accuracy and robustness of intrusion detection. This ensemble approach demonstrates superior performance in identifying potential security threats within cloud environments. To improve user interaction and testing, a user-friendly Flask framework with SQLite integration is introduced. This facilitates seamless signup and signin processes, ensuring practical usability in cybersecurity applications. The combined use of advanced ensemble techniques and a user-friendly interface contributes to the project's effectiveness in addressing security challenges within cloud-based intrusion detection systems.

ii) System Architecture:

The project architecture for "Deep Generative Learning Models for Cloud Intrusion Detection Systems" follows a systematic approach. It begins with the exploration and preprocessing of the dataset, crucial for effective model training. The dataset is then split into training and testing sets, laying the foundation for model evaluation. The core model, built for cloud intrusion detection, is extended with a Stacking Classifier incorporating SVM [28] for enhanced performance. The evaluation phase rigorously assesses the model's performance, considering key metrics. This comprehensive system architecture ensures a robust intrusion detection system tailored for cloud environments, with the ensemble model enhancing accuracy and adaptability to dynamic cyber threats, as validated through meticulous evaluation and analysis of its overall performance.

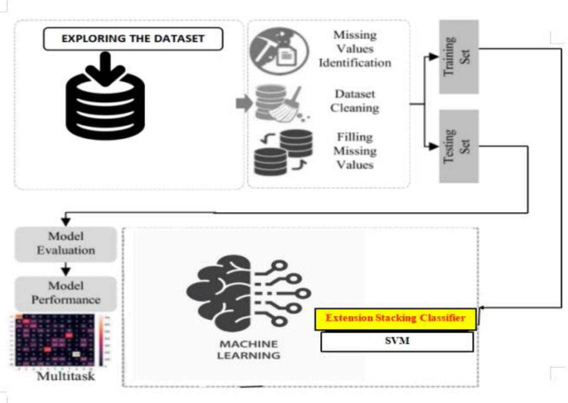


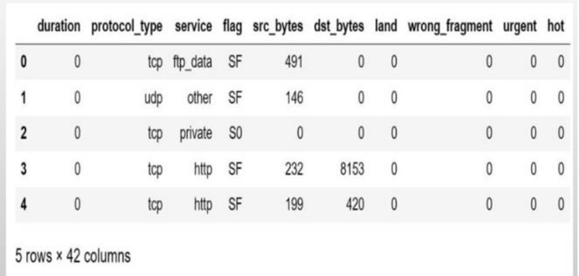
Fig 1 Proposed architecture

iii) Dataset collection:

This project involves getting familiar with the datasets used in the project. Understanding the structure, features, and properties of the KDD CUP dataset and UNSW NB15 [43] dataset is crucial for effective data preprocessing and subsequent model training.The reason for using these datasets is to demonstrate the effectiveness of the proposed solutions in detecting attacks.

KDD CUP DATASET

The KDD-CUP (Knowledge Discovery and Data Mining Cup) dataset is a widely used dataset for intrusion detection system research. In the context of deep generative learning models for cloud intrusion detection systems, the KDD-CUP dataset serves as a foundational dataset for training and evaluating models. Deep generative models can utilize this dataset to learn intricate patterns and features from the network traffic data, aiding in the development of more sophisticated intrusion detection systems for cloud environments.



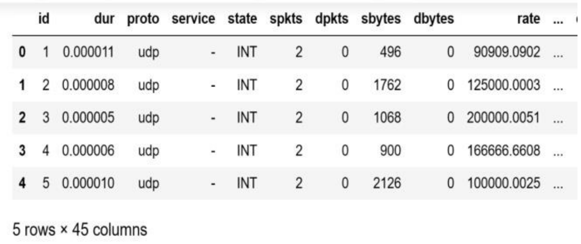
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

5 rows × 42 columns

Fig 2 KDD CUP dataset

UNSW-NB15 DATASET

The UNSW-NB15 dataset is a modern network traffic dataset designed to address some limitations of earlier datasets like KDD-CUP. In the context of deep generative learning models, the UNSW-NB15 dataset is essential for enhancing intrusion detection capabilities in cloud environments. The dataset provides a more up-to-date and relevant set of network traffic data, which can be leveraged by deep generative learning models to extract intricate features and patterns to better detect intrusions and cyber threats in the cloud.



	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...

5 rows × 45 columns

Fig 3 UNSB NB15 dataset

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

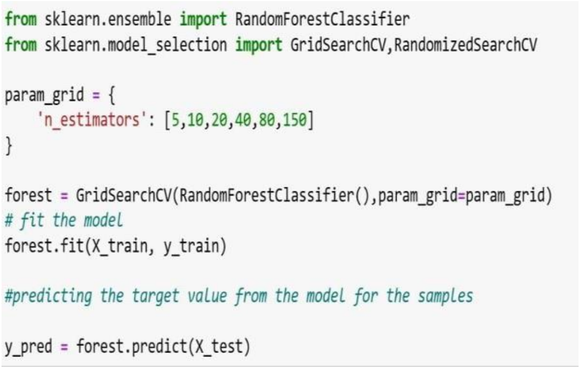
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

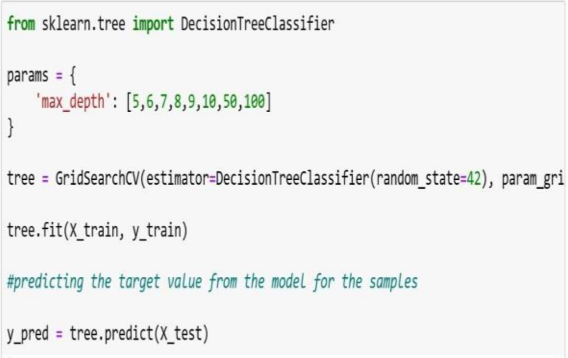
Random Forest is an ensemble of decision trees. It uses random subsets of data and features for each tree. The final prediction is a result of aggregating predictions from multiple trees. It's robust, accurate, and versatile for various machine learning tasks [4].



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV,RandomizedSearchCV

param_grid = {
    'n_estimators': [5,10,20,40,80,150]
}

forest = GridSearchCV(RandomForestClassifier(),param_grid=param_grid)
# fit the model
forest.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = forest.predict(X_test)
```

Fig 4 Random forest

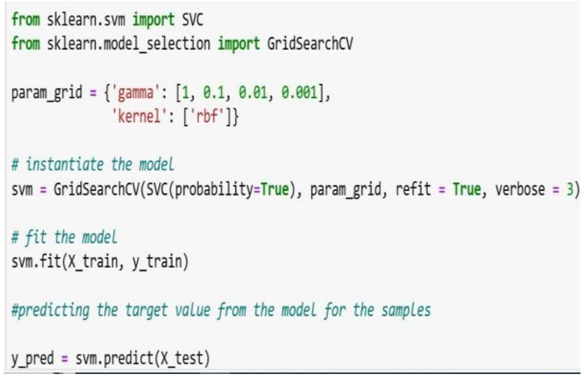
A decision tree is a flowchart-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome or prediction for a given input based on the feature values. It's a popular algorithm used for classification and regression tasks in machine learning [45].



```
from sklearn.tree import DecisionTreeClassifier

params = {
    'max_depth': [5,6,7,8,9,10,50,100]
}

tree = GridSearchCV(estimator=DecisionTreeClassifier(random_state=42), param_gri

tree.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = tree.predict(X_test)
```

Fig 5 Decision tree

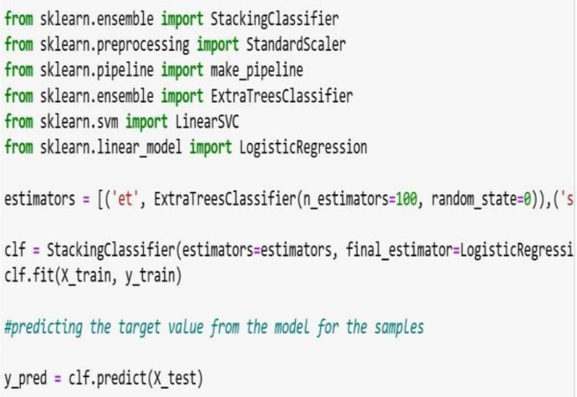
A **Support Vector Machine** (SVM) is a supervised learning algorithm that finds the optimal hyperplane to best separate data points of different classes, maximizing the margin and improving classification accuracy. [45]



```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

param_grid = {'gamma': [1, 0.1, 0.01, 0.001],
              'kernel': ['rbf']}

# instantiate the model
svm = GridSearchCV(SVC(probability=True), param_grid, refit = True, verbose = 3)

# fit the model
svm.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = svm.predict(X_test)
```

Fig 6 SVM

Stacking, also known as stacked generalization or stacking ensemble, is an ensemble learning technique that combines multiple machine learning models to improve predictive performance. It leverages the predictions of base models (level-0 models) as features to train a higher-level model (meta-model or level-1 model).



```
from sklearn.ensemble import StackingClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression

estimators = [('et', ExtraTreesClassifier(n_estimators=100, random_state=0)),('s

clf = StackingClassifier(estimators=estimators, final_estimator=LogisticRegressi
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = clf.predict(X_test)
```

Fig 7 Stacking

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

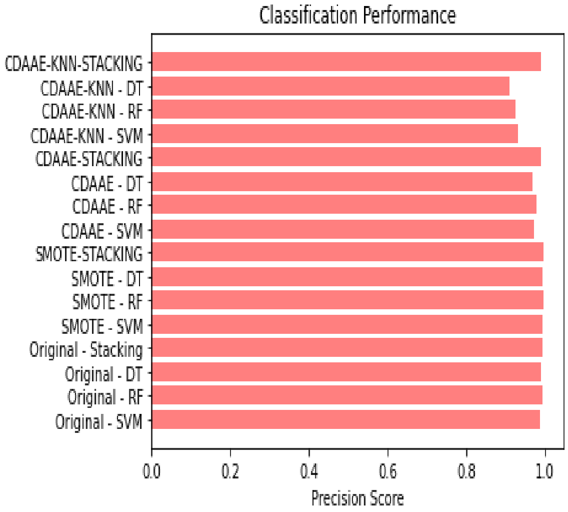


Fig 8 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

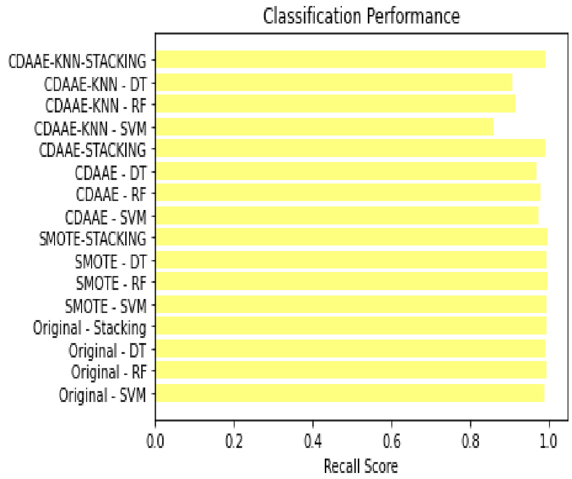


Fig 9 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

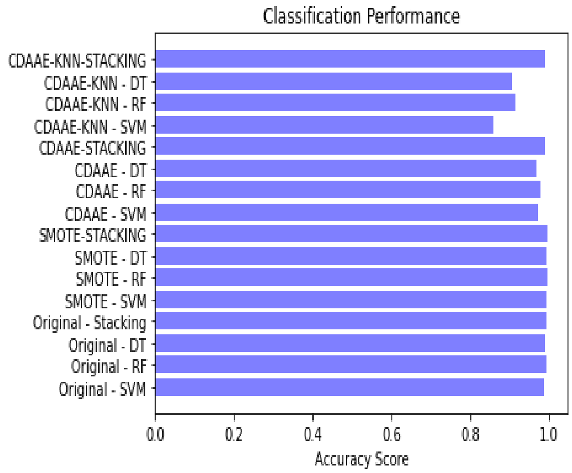


Fig 10 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

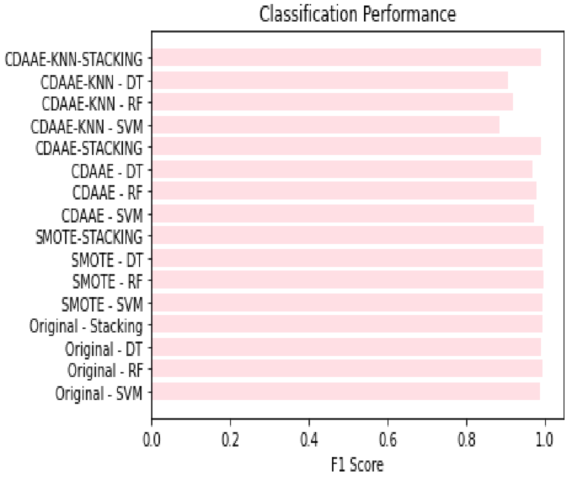
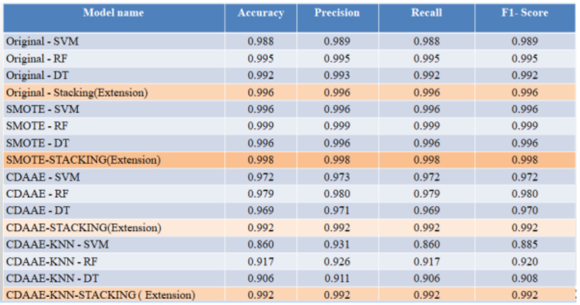


Fig 11 F1Score



Model name	Accuracy	Precision	Recall	F1- Score
Original - SVM	0.988	0.989	0.988	0.989
Original - RF	0.995	0.995	0.995	0.995
Original - DT	0.992	0.993	0.992	0.992
Original - Stacking(Extension)	0.996	0.996	0.996	0.996
SMOTE - SVM	0.996	0.996	0.996	0.996
SMOTE - RF	0.999	0.999	0.999	0.999
SMOTE - DT	0.996	0.996	0.996	0.996
SMOTE-STACKING(Extension)	0.998	0.998	0.998	0.998
CDAAE - SVM	0.972	0.973	0.972	0.972
CDAAE - RF	0.979	0.980	0.979	0.980
CDAAE - DT	0.969	0.971	0.969	0.970
CDAAE-STACKING(Extension)	0.992	0.992	0.992	0.992
CDAAE-KNN - SVM	0.860	0.931	0.860	0.885
CDAAE-KNN - RF	0.917	0.926	0.917	0.920
CDAAE-KNN - DT	0.906	0.911	0.906	0.908
CDAAE-KNN-STACKING (Extension)	0.992	0.992	0.992	0.992

Fig 12 Performance Evaluation

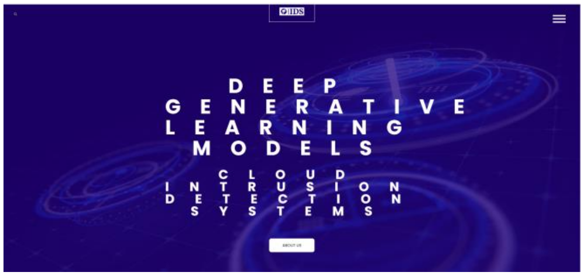


Fig 13 Home page

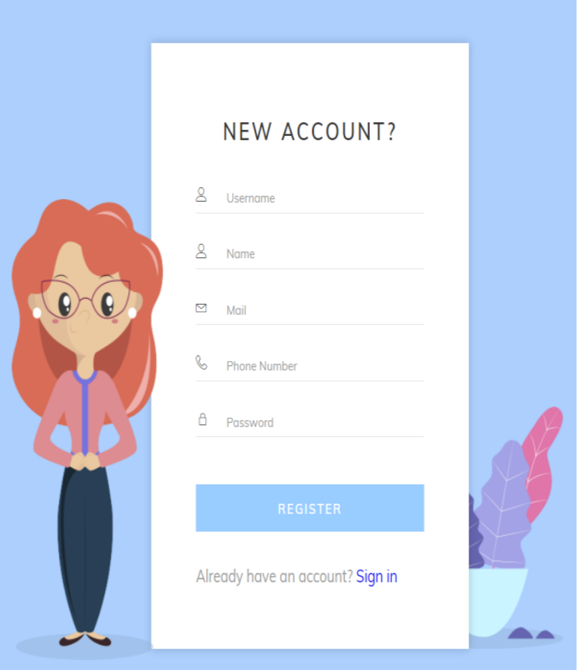
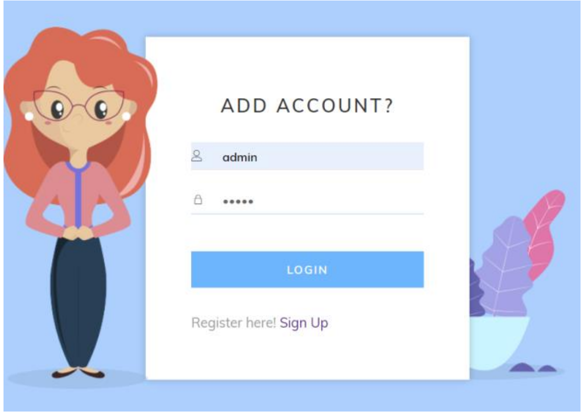


Fig 14 Signin page



admin

•••••

Fig 15 Login page

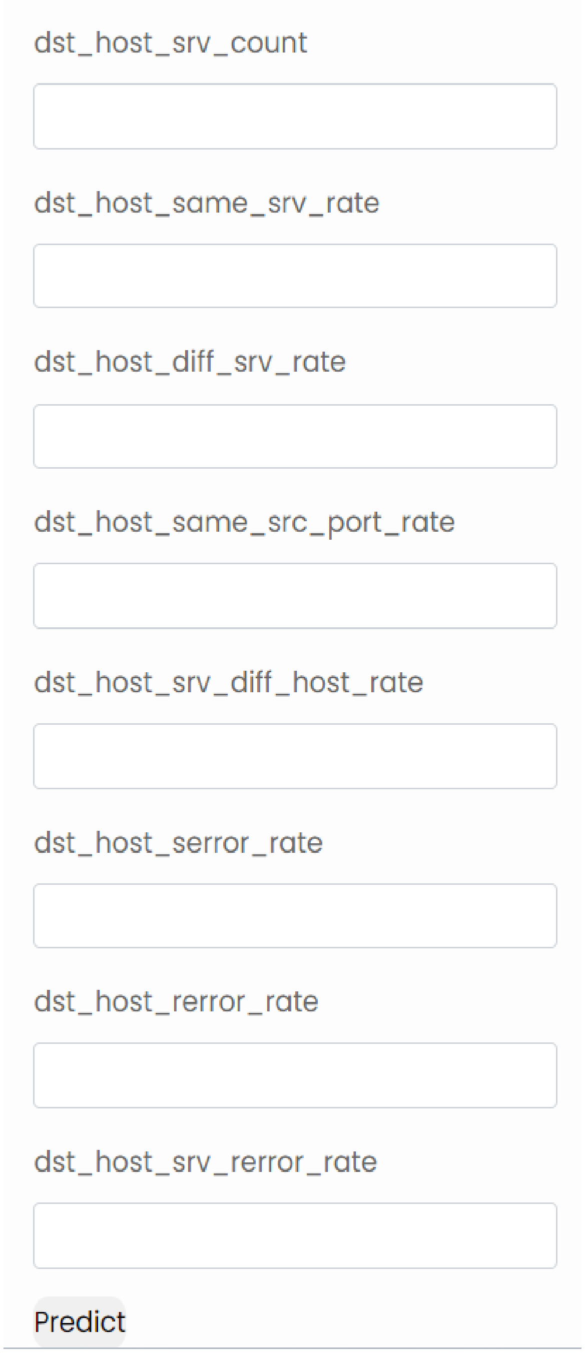
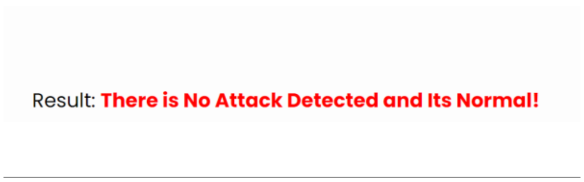


Fig 16User input



Result: **There is No Attack Detected and Its Normal!**

Fig 17 Predict result for given input

5. CONCLUSION

The CDAAE and CDAAE-KNN models serve as effective solutions for addressing data imbalance in Intrusion Detection System (IDS) datasets [42] within cloud environments. This is a crucial step in ensuring the accuracy and reliability of intrusion detection systems, particularly in cloud environments where diverse cyber threats may occur. These models, in particular, exhibit improved accuracy in detecting challenging Distributed Denial of Service (DDoS) attacks, including low-rate DDoS and application layer DDoS attacks [9, 19, 20]. Their effectiveness in identifying these sophisticated threats contributes to the overall robustness of the intrusion detection system, enhancing its capability to handle diverse and complex cyber threats. The project introduces ensemble techniques, such as the Stacking Classifier with Extratree Classifier + LinearSVC with LR, as an extension to the models. This ensemble approach demonstrates superior performance and robustness in intrusion detection. The diverse combination of classifiers within the ensemble enhances accuracy and adaptability, making it an effective solution for detecting intrusions in cloud environments. To enhance usability, the project incorporates a Flask-based user-friendly front end with secure authentication features. This ensures a practical and accessible solution for users interacting with the intrusion detection system. The integration of Flask and secure authentication adds a layer of robustness, making the system user-friendly while prioritizing data security, thus contributing to a comprehensive solution for intrusion detection in cloud environments.

6. FUTURE SCOPE

The future scope aims to further enhance the accuracy and effectiveness of cloud-based Intrusion Detection Systems (IDS) by employing advancements in deep generative learning models. This indicates a commitment to continuous improvement and refinement of the IDS for superior intrusion detection capabilities. Future efforts will focus on optimizing and fine-tuning the proposed models, CDAAE and CDAAE-KNN [13, 17]. This optimization will involve improving their ability to synthesize malicious samples, ultimately boosting the accuracy of the cloud IDS and making it more proficient in detecting a wide range of intrusions. The project envisions exploring additional deep learning techniques and algorithms beyond CDAAE and CDAAE-KNN. This exploration aims to broaden the horizons of the project by incorporating innovative methodologies to further enhance the detection and classification of unknown attacks within the cloud environment. The future scope emphasizes evaluating the proposed techniques on a larger and more diverse set of IDS datasets [11]. This validation process is essential for assessing the effectiveness and generalizability of the developed methods, ensuring that the project's contributions are applicable across a broad spectrum of real-world intrusion scenarios.

REFERENCES

[1] “Cloud Computing Market Worth $411 Billion by 2020.” [Online]. Available: https://www.itproportal.com/news/cloud-computing-marketworth-dollar411-billion-by-2020/ (Accessed: Jan. 20, 2019).

[2] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," IEEE Trans. Dependable Secure Comput., vol. 10, no. 4, pp. 212–224, Jul./Aug. 2013.

[3] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Learning latent representation for IoT anomaly detection," IEEE Trans. Cybern., early access, Sep. 18, 2020, doi: 10.1109/TCYB.2020.3013416.

[4] C. Yang, Z. Shi, H. Zhang, J. Wu, and X. Shi, "Multiple attacks detection in cyber-physical systems using random finite set theory," IEEE Trans. Cybern., vol. 50, no. 9, pp. 4066–4075, Sep. 2020.

[5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[6] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," IEEE Access, vol. 8, pp. 108346–108358, 2020.

[7] V. Borgiani, P. Moratori, J. F. Kazienko, E. R. R. Tubino, and S. E. Quincozes, "Toward a distributed approach for detection and mitigation of denial-of-service attacks within Industrial Internet of Things," IEEE Internet Things J., vol. 8, no. 6, pp. 4569–4578, Mar. 2021.

[8] N. Muraleedharan and B. Janet, " A deep learning based HTTP slow DoS classification approach using flow data," ICT Exp., vol. 7, no. 2, pp. 210–214, 2021.

[9] A. Praseed and P. S. Thilagam, "DDoS attacks at the application layer: Challenges and research perspectives for safeguarding Web applications," IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 661–685, 1st Quart., 2019.

[10] P. Mishra, E. S. Pilli, V. Varadharajan, and U. K. Tupakula, "Intrusion detection techniques in cloud environment: A survey," J. Netw. Comput. Appl., vol. 77, pp. 18–47, Jan. 2017.

[11] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," IEEE Trans. Cybern., vol. 49, no. 8, pp. 3074–3087, Aug. 2019.

[12] A. D. Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi, "Racing for Unbalanced Methods Selection," in Proc. 14th Int. Conf. Intell. Data Eng. Autom. Learn., vol. 8206, 2013, pp. 24–31. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-41278-3_4

[13] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, arXiv:1511.05644.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, 2002.

[15] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 2, pp. 539–550, Apr. 2009.

[16] H. Chen, Z. Liu, Z. Liu, and P. Zhang, "ACGAN-based data augmentation integrated with long-term scalogram for acoustic scene classification," 2020, arXiv:2005.13146.

[17] M. Sadeghi, S. Leglaive, X. Alameda-Pineda, L. Girin, and R. Horaud, "Audio-visual speech enhancement using conditional variational auto-encoders," IEEE/ACM Trans. Audio, Speech, Language Process., vol. 28, pp. 1788–1800, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9110765

[18] A. Karami, "An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities," Expert Syst. Appl., vol. 108, pp. 36–60, Oct. 2018.

[19] A. S. Khader, D. Lai, Y. Li, and M. Diykh, "An efficient DDoS TCP flood attack detection and prevention system in a cloud environment," IEEE Access, vol. 5, pp. 6036–6048, 2017.

[20] B. Kiranmai and A. Damodaram, "Extenuate DDoS attacks in cloud," in Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT), 2016, pp. 235–238.

[21] M. Chouhan and H. Hasbullah, "Adaptive detection technique for cachebased side channel attack using bloom filter for secure cloud," in Proc. 3rd Int. Conf. Comput. Inf. Sci. (ICCOINS), 2016, pp. 293–297.

[22] K. Wang and Y. Hou, "Detection method of SQL injection attack in cloud computing environment," in Proc. IEEE Adv. Inf. Manage. Commun. Electron. Autom. Control Conf. (IMCEC), 2016, pp. 487–493.

[23] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. N. Nguyen, and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," in Proc. IEEE Wireless Commun. Netw. Conf., 2018, pp. 1–6.

[24] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," Mobile Netw. Appl., vol. 21, no. 3, pp. 494–505, 2016.

[25] S. Dey, Y. Qiang, and S. Srinivas, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," Inf. Fusion, vol. 49, pp. 205–215, Sep. 2019.

[26] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in Proc. IEEE Int. Joint Conf. Neural Netw., 2016, pp. 4368–4374.

[27] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[28] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," Int. J. Knowl. Eng. Soft Data Paradigms, vol. 3, no. 1, pp. 4–21, 2011.

[29] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," 2013, arXiv:1305.1707.

[30] D. Yuan et al., "Intrusion detection for smart home security based on data augmentation with edge

computing," in Proc. IEEE Int. Conf. Commun. (ICC), 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148632.

[31] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward effective intrusion detection using log-cosh conditional variational autoencoder," IEEE Internet Things J., vol. 8, no. 8, pp. 6187–6196, Apr. 2021, doi: 10.1109/JIOT.2020.3034621.

[32] Q. Yu and W. Lam, "Data augmentation based on adversarial autoencoder handling imbalance for learning to rank" in Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell., Jan. 2019, pp. 411–418.

[33] I. Goodfellow et al., "Generative adversarial nets," in Advances in Neural Information Processing Systems, vol. 27. Red Hook, NY, USA: Curran Assoc., 2014, pp. 2672–2680.

[34] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Proc. Annu. Conf. Neural Inf. Process. Syst., Barcelona, Spain, 2016, pp. 2226–2234.

[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, arXiv:1312.6114.

[36] A. Creswell and A. A. Bharath, "Denoising adversarial autoencoders," IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 4, pp. 968–984, Apr. 2019.

[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1096–1103.

[38] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new oversampling method in imbalanced data sets learning," in Proc. Int. Conf. Intell. Comput., 2005, pp. 878–887.

[39] J. Cervantes, F. García-Lamont, L. Rodríguez-Mazahua, A. López Chau, J. S. R. Castilla, and A. Trueba, "PSO-based method for SVM classification on skewed data sets," Neurocomputing, vol. 228, pp. 187–197, Mar. 2017.

[40] R. Kumar, S. P. Lal, and A. Sharma, "Detecting denial of service attacks in the cloud," in Proc. IEEE 14th Int. Conf. Dependable Auton. Secure Comput. 14th Int. Conf. Pervasive Intell. Comput. 2nd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech), 2016, pp. 309–316.

[41] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP), Jan. 2018, pp. 1–9.

[42] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Security Defense Appl., 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[43] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), 2015, pp. 1–6.

[44] M. Feurer and F. Hutter, "Hyperparameter optimization," in Automated Machine Learning (The

Springer Series on Challenges in Machine Learning), F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Cham, Switzerland: Springer, 2019, pp. 3–33. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-05318-5_1#citeas

[45] "Sklearn Tutorial." [Online]. Available: http://scikit-learn.org/stable/ (Accessed: Apr. 24, 2018).

[46] D. Powers, "Evaluation: From precision, recall and F measure to ROC, informedness, markedness and correlation," J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

[47] M. Bekkar, H. Djema, and T. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," J. Inf. Eng. Appl., vol. 3, no. 10, pp. 27–38, 2013.

[48] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Nonparallel voice conversion with auxiliary classifier variational autoencoder," IEEE/ACM Trans. Audio, Speech, Language Process., vol. 27, no. 9, pp. 1432–1443, Sep. 2019