Intrusion Detection System using Neural Networks

Made by

Krish Chatterjie - 20BCE0516
Yajat Malhotra - 20BCE0554
Ishita Chauhan - 20BCE2109

Table of Contents

1. Abstract
2. Introduction
3. Related Works
4. References
5. Proposed Work
6. Implementation
7. Results
8. Practical Aspects
9. What we learnt
10. Conclusion

Abstract

- An intrusion detection system, often known as an IDS, is now a standard component of every cutting-edge information and communications technology (ICT) system because of the growing concern for online safety in everyday life.
- IDS is calling for the requirement of integrating Neural Networks because of a number of reasons, some of which include the lack of assurance in determining the sorts of assaults and the rising complexity of modern cyberattacks.
- In this project we have compared between multiple different neural networks

Abstract

- We compared the following three neural networks:
  - Shallow Neural Network
  - Deep Neural Network
  - Convolutional Neural Network
- We found out the accuracy of the three models and compared them

Introduction

- Due to a need for cyber safety in the everyday world, intrusion detection systems (IDS) have evolved into a crucial layer in all modern ICT systems.
- In order to manage and keep track of network traffic packets at various places for any potential intrusions or anomalies, a network intrusion detection system (NIDS) may be composed of both hardware (sensors) and software (console).
- IDS calls for the inclusion of Deep Neural Networks (DNNs) because to factors such as the difficulty in identifying attack types and the rising complexity of modern cyberattacks.

Introduction

Intrusion Detection Systems

- This system watches network traffic for suspicious activity and sends out alerts when it is found.
- It is software that checks a system or network for malicious activities or policy violations.
- Any illegal activity or violation is often recorded either centrally using a security information and event management (SIEM) system or notified to an administrator.
- A SIEM system combines outputs from several sources and employs alarm filtering methods to distinguish between legitimate and erroneous alarms.

Introduction

Deep Learning

- Deep learning makes an attempt to emulate the capabilities of the human brain, allowing systems to cluster data and produce predictions that are incredibly accurate.
- It does so on the basis of data inputs, weight and biases.
- Together, these components accurately identify, categorise, and characterise items in the data.
- Additional hidden layers can help to tune and refine for accuracy even if a neural network with only one layer can still make approximation predictions.

Introduction

Convolutional Neural Networks

- In deep learning, a convolutional neural network (CNN) is a class of artificial neural network (ANN).
- It consists of an input layer, hidden layers and an output layer. The hidden layers include layers that perform convolutions.
- They provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image.
- CNNs can be computationally demanding, requiring graphical processing units (GPUs) to train models.

Introduction

Shallow and Deep Neural Networks

- Shallow neural networks is a term used to describe neural networks that usually have only one hidden layer.
- As opposed to deep neural networks which have several hidden layers, often of various types.
- There are papers that highlight that deep neural networks with the right architectures achieve better results than shallow ones that have the same computational power (e.g. number of neurons or connections).
- The main explanation is that the deep models are able to extract/build better features than shallow models and to achieve this they are using the intermediate hidden layers.

Related Works

1. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security

- In this study, DNNs were used to foresee Network Intrusion Detection System attacks (N-IDS).
- The KDDCup-'99 dataset has been utilised for training and benchmarking the network, and a DNN with a learning rate of 0.1 is applied and run for 1000 epochs.
- For comparison, the same dataset is trained using a variety of other traditional machine learning algorithms and DNN with layers ranging from 1 to 5.
- Comparing the findings, it was found that a DNN with three layers outperformed all other traditional machine learning techniques.

Related Works

2. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks

- This research explores deep learning models for intrusion detection systems and proposes a deep learning method for intrusion detection utilising recurrent neural networks (RNN-IDS).
- Additionally, the model's performance in binary classification and multiclass classification is investigated, as well as how the number of neurons and various learning rates affect the performance of the suggested model.
- The experimental results demonstrate that RNN-IDS performs better than typical machine learning classification methods in both binary and multiclass classification, and that it is particularly well suited for developing a classification model with high accuracy.
- The RNN-IDS model enhances intrusion detection accuracy and offers a fresh approach to intrusion detection research.

Related Works

3. Method of Intrusion Detection using Deep Neural Network

- The DNN model, a deep learning technique, is studied in this work as part of an artificial intelligence intrusion detection system for efficient assault detection.
- For testing and training, it made use of the well-known KDD Cup 99 dataset for intrusion detection.
- To achieve the goals of the study, test data were produced through data preprocessing and sample extraction. The training set was a dataset with 10% of the corrected data, and the testing set was the whole dataset, which contained roughly 4.9 million entries.
- The outcomes demonstrate a 99% average accuracy and detection rate. Additionally, the false alarm rate was 0.08%, indicating that there is extremely little chance that legitimate data will be mistakenly identified as an assault.
- However, in order to combat distributed denial of service (DDoS) attacks, time series data analysis utilising the recurrent neural network (RNN) model and the long short-term memory (LSTM) model will be necessary.

Related Works

4. Practical real-time intrusion detection using machine learning approaches

- In this research, they suggest a supervised machine learning method for real-time intrusion detection.
- Their method is straightforward and effective, and it works with a variety of machine learning strategies.
- They used a variety of well-known machine learning techniques to assess the effectiveness of our IDS strategy.
- The Decision Tree technique can outperform the other techniques, according to the findings of their experiments.
- In order to categorise online network data as normal or attack data, they further built a real-time intrusion detection system (RT-IDS) using the Decision Tree technique.
- In addition, using information gain as the feature selection criterion, they discovered 12 crucial characteristics of network data that are pertinent to identifying network attacks.

Related Works

5. A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection

- In order to determine the root of issues with various machine learning techniques in identifying invasive behaviours, a thorough research and analysis of numerous machine learning techniques has been conducted in this work.
- According to each attack, an attack classification and feature mapping is given.
- Also, problems with utilising network attack datasets to detect low-frequency assaults are examined, and workable solutions are offered.
- Machine learning algorithms (Decision Tree, Neural Network, Naive Bayes, Support Vector Machine and Fuzzy Association rules) have been examined and contrasted in terms of how well they can identify the different types of attacks.
- Additionally, each category's drawbacks are mentioned.

Related Works

6. Intrusion Detection Using Machine Learning: A Comparison Study

- By examining the combinations of the majority of the widely used feature selection techniques and classifiers, this research suggests an IDS for networks that uses machine learning and has a good union of feature selection technique and classifier.
- Using feature selection approaches, a subset of critical features is chosen from the initial collection of features, and the subset is then used to train several classifiers to create the IDS.
- On the NSL-KDD dataset, five folds cross validation is used to find results.
- Finally, it is discovered that the K-NN classifier performs better than the competition, and that the information gain ratio-based feature selection method is superior than the others.

Related Works

7. Towards Efficient Intrusion Detection using Deep Learning Techniques: A Review

- The review organises the Deep Learning models and looks at 23 publications that successfully apply Deep Learning to intrusion detection systems.
- The article has looked at different Deep Learning Models to help with malware and unwanted traffic detection.
- It categorises the approach taken to spot the intrusion in every instance.
- The classification shows that, with high accuracy percentages, autoencoders and recurrent neural networks beat CNN-based models.
- This makes sense given that CNN are primarily intended for use in image processing applications.

Related Works

8. A Deep Learning Approach for Network Intrusion Detection System

- System administrators can identify network security breaches in their businesses with the aid of a Network Intrusion Detection System (NIDS).
- However, there are numerous obstacles to overcome when creating a versatile and effective NIDS for unanticipated and unpredictable attacks.
- We provide a deep learning-based methodology for creating such an effective and adaptable NIDS.
- On NSL-KDD, a benchmark dataset for network intrusion, we employ Self-taught Learning (STL), a deep learning-based technique.

Related Works

9. Classification Model for Accuracy and Intrusion Detection using Machine Learning Approach

- In this study, the accuracy and processing time of three different classification machine learning algorithms—Naive Bayes (NB), Support Vector Machine (SVM), and K-nearest neighbour (KNN)—were compared on the UNSW-NB15 dataset in order to determine which algorithm was most effective at learning the pattern of suspicious network activity.
- The information obtained from the feature set comparison was then used as data feeds to train the system for future intrusion behaviour prediction and analysis using the best-fit algorithm selected from the previously mentioned three algorithms based on the performance metrics discovered.
- Additionally, the confusion matrix, classification reports (Precision, Recall, and F1-score), and support-validation status discovered during the testing phase of the model utilised in this approach were generated and compared.

Related Works

10. A Deep Learning Approach to Network Intrusion Detection

- These issues are addressed by the unique deep learning technique for intrusion detection presented in this paper.
- They describe their nonsymmetric deep autoencoder (NDAE) proposal for unsupervised feature learning.
- They also offer our original deep learning categorization model, which was built using stacked NDAEs.
- Utilizing the benchmark KDD Cup '99 and NSL-KDD datasets, their suggested classifier has been implemented in TensorFlow that is GPU-enabled.

References

1. Vigneswaran, Rahul K., et al. "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security." *2018 9th International conference on computing, communication and networking technologies (ICCCNT).* IEEE, 2018.
2. Yin, Chuanlong, et al. "A deep learning approach for intrusion detection using recurrent neural networks." *IEEE Access 5* (2017): 21954-21961.
3. Kim, Jin, et al. "Method of intrusion detection using deep neural network." *2017 IEEE international conference on big data and smart computing (BigComp).* IEEE, 2017.
4. Sangkatsanee, Phurivit, Naruemon Wattanapongsakorn, and Chalermpol Charnsripinyo. "Practical real-time intrusion detection using machine learning approaches." *Computer Communications 34.18 (2011): 2227-2235.*
5. Mishra, Preeti, et al. "A detailed investigation and analysis of using machine learning techniques for intrusion detection." *IEEE communications surveys & tutorials 21.1 (2018): 686-728.*

References

6. Biswas, Saroj Kr. "Intrusion detection using machine learning: A comparison study." *International Journal of pure and applied mathematics* 118.19 (2018): 101-114.
7. Vani, R. "Towards efficient intrusion detection using deep learning techniques: a review." *Int J Adv Res Comput Commun Eng ISO 3297* (2017): 2007.
8. Javaid, Ahmad, et al. "A deep learning approach for network intrusion detection system." *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS).* 2016.
9. Agarwal, Arushi, et al. "Classification model for accuracy and intrusion detection using machine learning approach." *PeerJ Computer Science 7* (2021): e437.
10. Shone, Nathan, et al. "A deep learning approach to network intrusion detection." *IEEE transactions on emerging topics in computational intelligence 2.1* (2018): 41-50.

Proposed Work

Tools Used

- Python
- Tensorflow
- Keras

Proposed Work

Data Set (KDD-99 Benchmark dataset)

- This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining.
- The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections.
- This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.
- Link: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Proposed Work

Data Features

- 'duration', 'protocol_type', 'flag', 'src_bytes', 'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot', 'num_failed_logins', 'logged_in', 'num_compromised', 'root_shell', 'su_attempted', 'num_file_creations', 'num_shells', 'num_access_files', 'is_guest_login', 'count', 'srv_count', 'serror_rate', 'rerror_rate', 'same_srv_rate', 'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'Attack Type'

Proposed Work

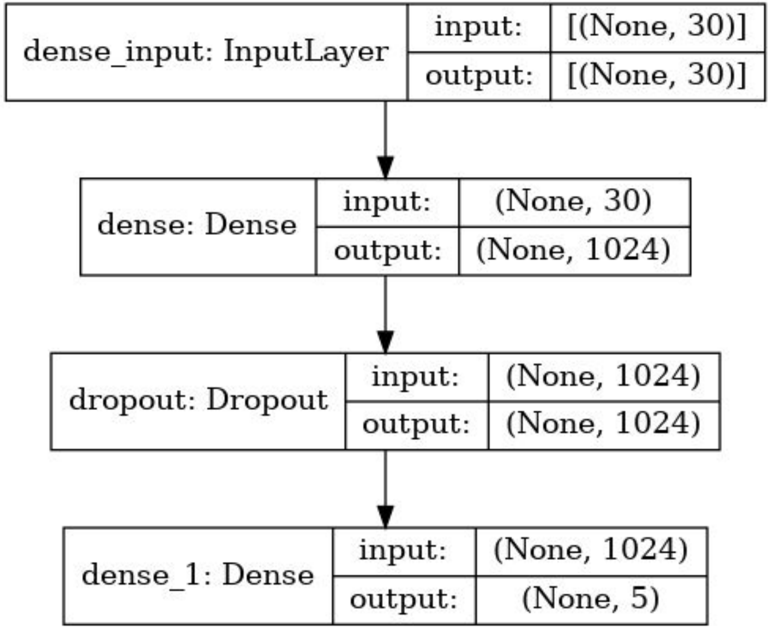
Model Used

- 3 types of models where made and compared:
  - Shallow Neural Network
  - Deep Neural Network
  - Convolutional Neural Network
- Then the results and accuracy of the three were compared to find out the best model out of the three.

Proposed Work

Shallow Neural Network

- Consists of 1 hidden layer
- Consists of 1 dropout layer to prevent overfitting
- The output layer classifies the input into 5 different classes
- Adapted from Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security | IEEE Conference Publication | IEEE Xplore



dense_input: InputLayer | input: [(None, 30)] | output: [(None, 30)]

dense: Dense | input: (None, 30) | output: (None, 1024)

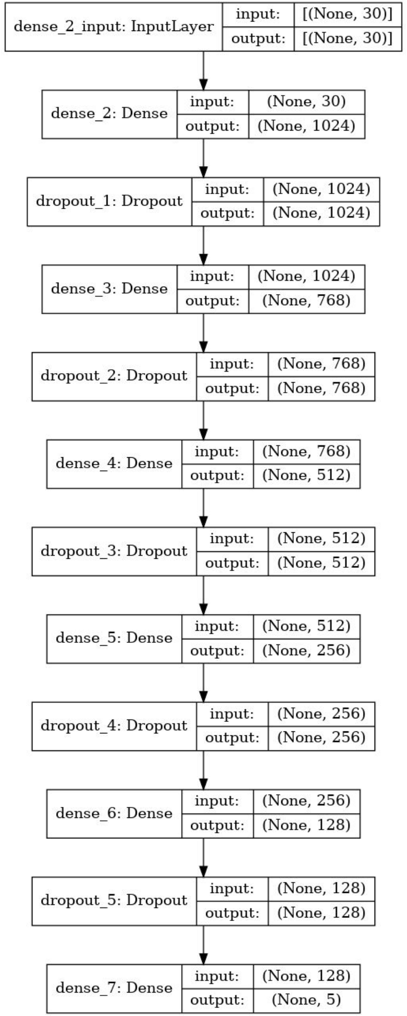
dropout: Dropout | input: (None, 1024) | output: (None, 1024)

dense_1: Dense | input: (None, 1024) | output: (None, 5)

Proposed Work

Deep Neural Network

- Consists of 5 hidden layers with varying number of nodes
- Consists of 5 dropout layers to prevent overfitting
- The output layer classifies the input into 5 different classes
- Adapted from Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security | IEEE Conference Publication | IEEE Xplore



dense_2_input: InputLayer | input: [(None, 30)] | output: [(None, 30)]

dense_2: Dense | input: (None, 30) | output: (None, 1024)

dropout_1: Dropout | input: (None, 1024) | output: (None, 1024)

dense_3: Dense | input: (None, 1024) | output: (None, 768)

dropout_2: Dropout | input: (None, 768) | output: (None, 768)

dense_4: Dense | input: (None, 768) | output: (None, 512)

dropout_3: Dropout | input: (None, 512) | output: (None, 512)

dense_5: Dense | input: (None, 512) | output: (None, 256)

dropout_4: Dropout | input: (None, 256) | output: (None, 256)

dense_6: Dense | input: (None, 256) | output: (None, 128)

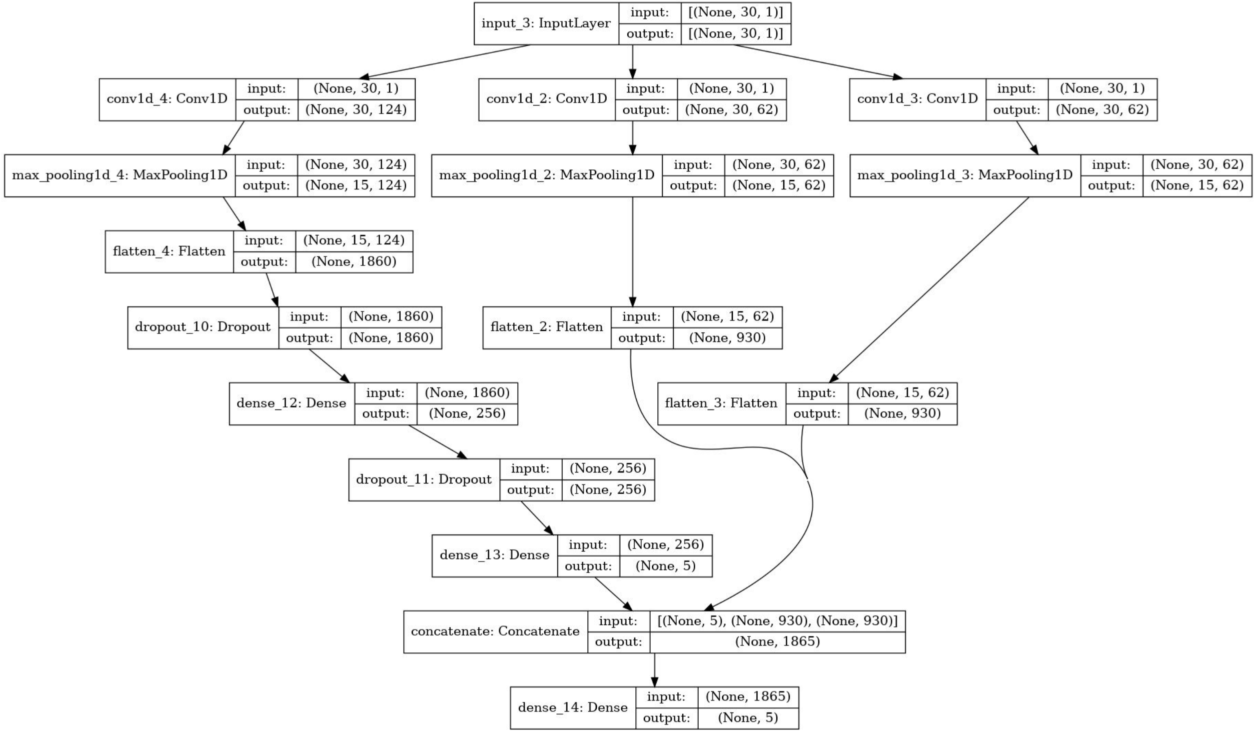
dropout_5: Dropout | input: (None, 128) | output: (None, 128)

dense_7: Dense | input: (None, 128) | output: (None, 5)

Proposed Work

Convolutional Neural Network

- Adapted from Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks | Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence (acm.org)



input_3: InputLayer — input: [(None, 30, 1)] — output: [(None, 30, 1)]

conv1d_4: Conv1D — input: (None, 30, 1) — output: (None, 30, 124)

conv1d_2: Conv1D — input: (None, 30, 1) — output: (None, 30, 62)

conv1d_3: Conv1D — input: (None, 30, 1) — output: (None, 30, 62)

max_pooling1d_4: MaxPooling1D — input: (None, 30, 124) — output: (None, 15, 124)

max_pooling1d_2: MaxPooling1D — input: (None, 30, 62) — output: (None, 15, 62)

max_pooling1d_3: MaxPooling1D — input: (None, 30, 62) — output: (None, 15, 62)

flatten_4: Flatten — input: (None, 15, 124) — output: (None, 1860)

dropout_10: Dropout — input: (None, 1860) — output: (None, 1860)

flatten_2: Flatten — input: (None, 15, 62) — output: (None, 930)

dense_12: Dense — input: (None, 1860) — output: (None, 256)

flatten_3: Flatten — input: (None, 15, 62) — output: (None, 930)

dropout_11: Dropout — input: (None, 256) — output: (None, 256)

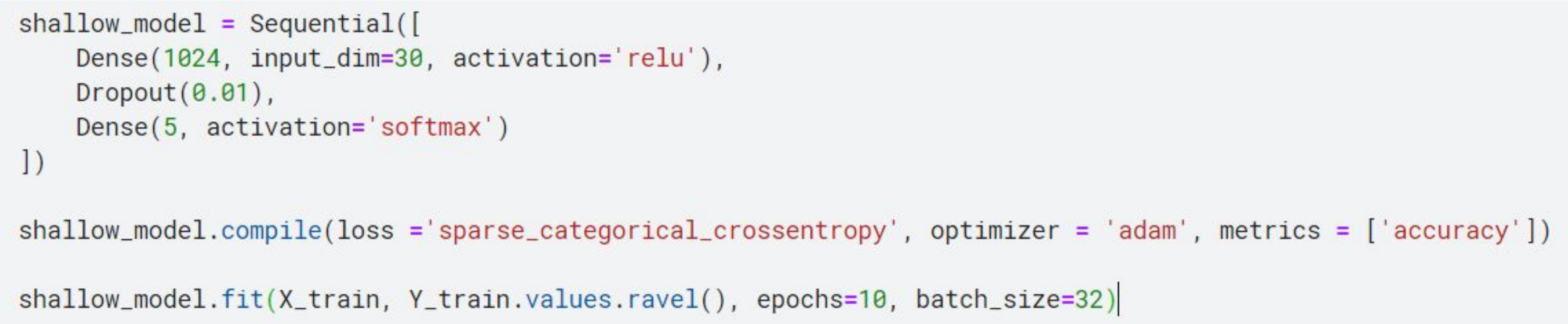
dense_13: Dense — input: (None, 256) — output: (None, 5)

concatenate: Concatenate — input: [(None, 5), (None, 930), (None, 930)] — output: (None, 1865)

dense_14: Dense — input: (None, 1865) — output: (None, 5)

Implementation

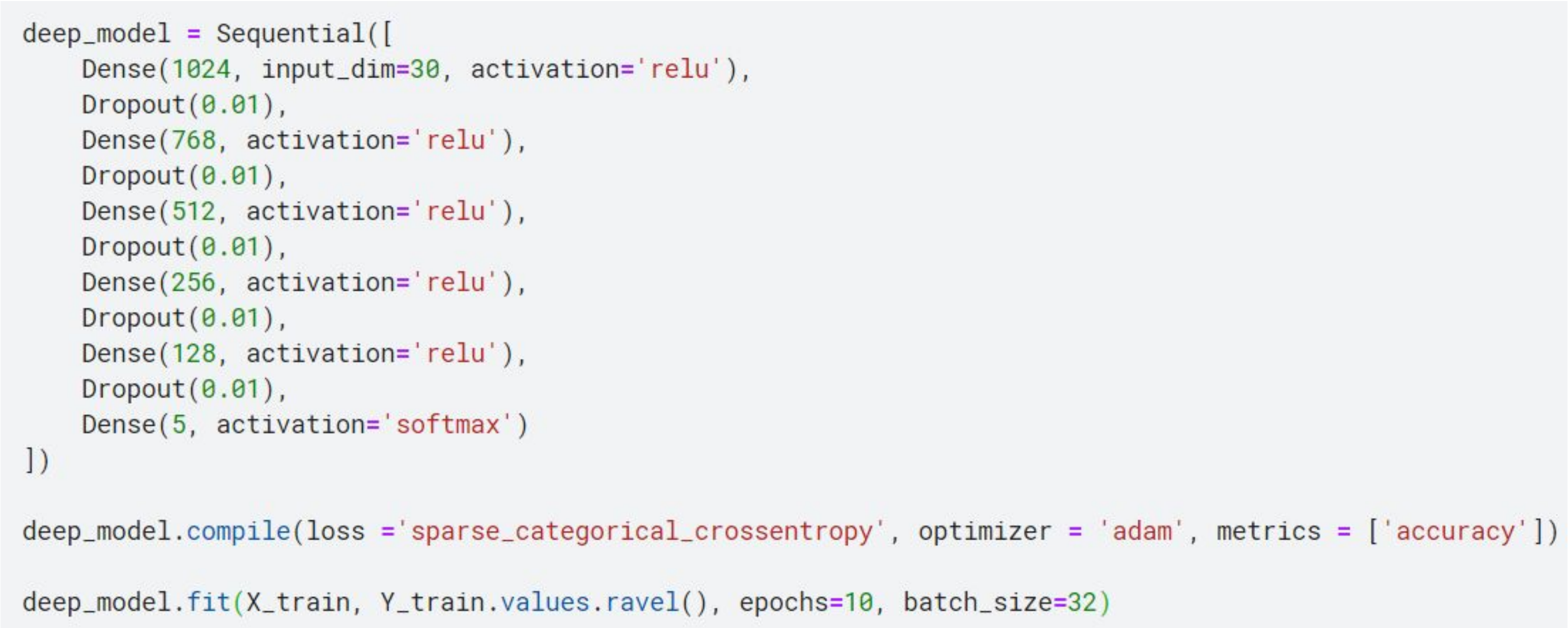
Shallow Neural Network



```
shallow_model = Sequential([
    Dense(1024, input_dim=30, activation='relu'),
    Dropout(0.01),
    Dense(5, activation='softmax')
])

shallow_model.compile(loss ='sparse_categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

shallow_model.fit(X_train, Y_train.values.ravel(), epochs=10, batch_size=32)
```

Implementation

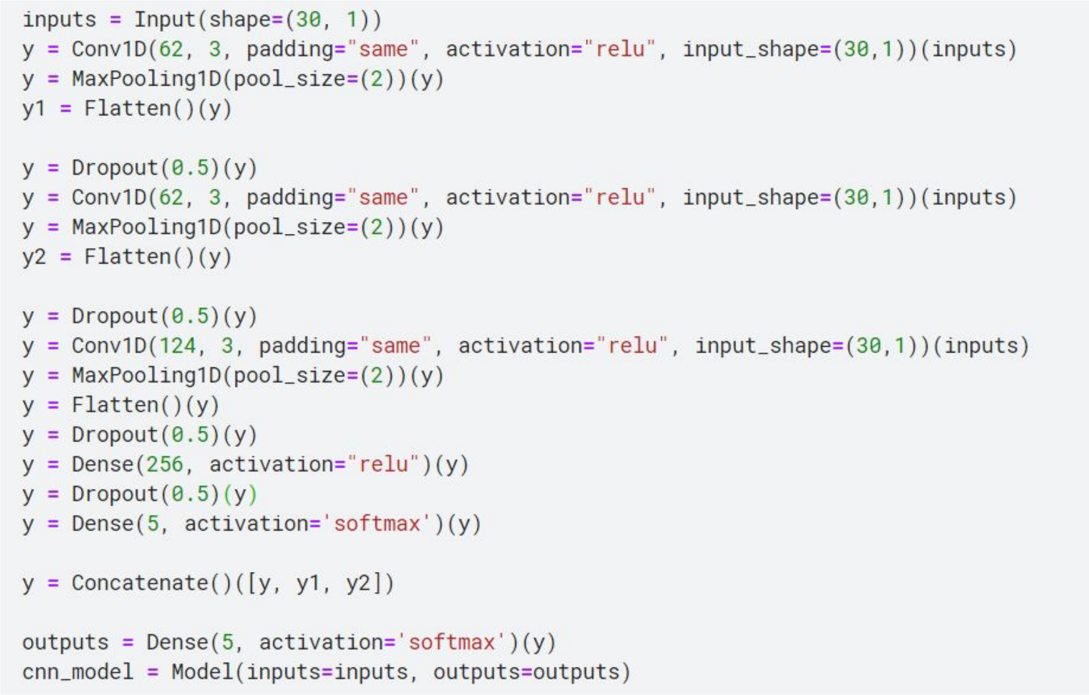
Deep Neural Network



```
deep_model = Sequential([
    Dense(1024, input_dim=30, activation='relu'),
    Dropout(0.01),
    Dense(768, activation='relu'),
    Dropout(0.01),
    Dense(512, activation='relu'),
    Dropout(0.01),
    Dense(256, activation='relu'),
    Dropout(0.01),
    Dense(128, activation='relu'),
    Dropout(0.01),
    Dense(5, activation='softmax')
])

deep_model.compile(loss ='sparse_categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

deep_model.fit(X_train, Y_train.values.ravel(), epochs=10, batch_size=32)
```

Implementation

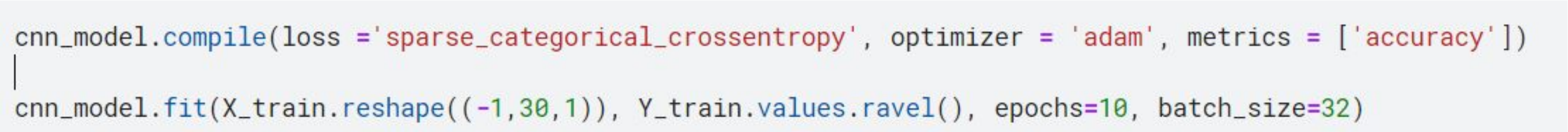
Convolutional Neural Network



```
inputs = Input(shape=(30, 1))
y = Conv1D(62, 3, padding="same", activation="relu", input_shape=(30,1))(inputs)
y = MaxPooling1D(pool_size=(2))(y)
y1 = Flatten()(y)

y = Dropout(0.5)(y)
y = Conv1D(62, 3, padding="same", activation="relu", input_shape=(30,1))(inputs)
y = MaxPooling1D(pool_size=(2))(y)
y2 = Flatten()(y)

y = Dropout(0.5)(y)
y = Conv1D(124, 3, padding="same", activation="relu", input_shape=(30,1))(inputs)
y = MaxPooling1D(pool_size=(2))(y)
y = Flatten()(y)
y = Dropout(0.5)(y)
y = Dense(256, activation="relu")(y)
y = Dropout(0.5)(y)
y = Dense(5, activation='softmax')(y)

y = Concatenate()([y, y1, y2])

outputs = Dense(5, activation='softmax')(y)
cnn_model = Model(inputs=inputs, outputs=outputs)
```

Implementation

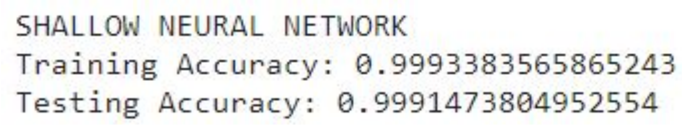
Convolutional Neural Network (Cont…)



```
cnn_model.compile(loss ='sparse_categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

cnn_model.fit(X_train.reshape((-1,30,1)), Y_train.values.ravel(), epochs=10, batch_size=32)
```

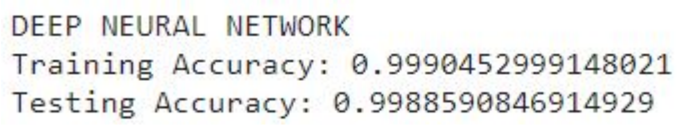
Results

Shallow Neural Network



```
SHALLOW NEURAL NETWORK
Training Accuracy: 0.9993383565865243
Testing Accuracy: 0.9991473804952554
```

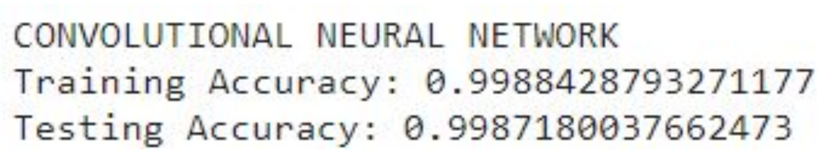
Deep Neural Network



```
DEEP NEURAL NETWORK
Training Accuracy: 0.9990452999148021
Testing Accuracy: 0.9988590846914929
```

Results

Convolutional Neural Network



```
CONVOLUTIONAL NEURAL NETWORK
Training Accuracy: 0.9988428793271177
Testing Accuracy: 0.9987180037662473
```

- All of the neural network has comparative results
- Out of these, the shallow neural network seemed to have worked the best.
- As the network is shallow, it is less complex and performs faster as well.

Practical Aspects

- This can be used to detect and take measures before an intrusion happens
- It checks a system or network for malicious activities or policy violations.
- It is more reliable than a rule based Intrusion Detection System.
- It can adapt to newer malicious activities easily and detect them.

What we learnt

- We learnt what all parameters an Intrusion Detection System works on.
- We implemented 3 different types of neural networks:
  - Shallow Neural Network
  - Deep Neural Network
  - Convolutional Neural Network
- We compared the different neural networks to find which one works the best.

Conclusion

- An intrusion detection system, often known as an IDS, is now a standard component of every cutting-edge information and communications technology (ICT) system because of the growing concern for online safety in everyday life.
- Neural Networks in general work really well as an IDS.
- Increasing the complexity of the neural network has not much accuracy improvements.
- Smaller the network, faster it works!