



Saveetha Institute of Medical and Technical Sciences
Saveetha School of Engineering



CAPSTONE PROJECT

COURSE CODE: CSA4715

COURSE NAME: Deep learning for Neural Network

Project Title

**UNSUPERVISED ANOMALY DETECTION CAPTURING COMPLEX
PATTERNS**

Submitted by:

T.Thanusree (192224193)

GS. Navamitha (192124163)

Guided by

Dr. POONGAVANAM N,

Associate Professor,

Department of Artificial Intelligence and Data Science

ABSTARCT

Anomaly detection, also known as outlier detection, is the process of identifying patterns or instances in data that do not conform to expected behavior. The goal is to distinguish anomalies from normal data patterns. This paper provides a comprehensive review of anomaly detection techniques, methodologies, and applications. Beginning with an overview of the concept and significance of anomaly detection, it delves into various approaches including statistical methods, machine learning algorithms, and deep learning architectures. The review encompasses both traditional techniques such as Gaussian distribution models and novel approaches like deep autoencoders and adversarial learning. Additionally, the paper discusses the challenges associated with anomaly detection such as imbalanced data, interpretability, and scalability, along with potential solutions proposed in recent research. Furthermore, it highlights real-world applications of anomaly detection across different domains, showcasing its effectiveness in fraud detection, fault diagnosis, intrusion detection, and beyond. Finally, the paper outlines future directions and emerging trends in anomaly detection research, emphasizing the integration of multi-modal data sources, reinforcement learning techniques, and the development of robust, scalable solutions to tackle evolving threats and challenges.

KEYWORDS: Deep autoencoders , Machine learning algorithms , Gaussian distribution models .

Chapter 1

Introduction

Anomaly detection, a critical component in data analysis and surveillance systems, serves as a sentinel against deviations from the norm. In an era where data is generated at an unprecedented pace across diverse domains such as finance, cybersecurity, healthcare, and industrial processes, the ability to identify anomalies swiftly and accurately has become paramount. Anomalies, often indicative of critical events, malicious activities, or faults, can pose significant risks if left undetected. Consequently, anomaly detection techniques have garnered increasing attention from researchers and practitioners alike.

The essence of anomaly detection lies in discerning patterns and deviations that stray from expected behavior within datasets. Traditional methods, rooted in statistical principles, leverage measures of central tendency and dispersion to identify outliers. However, with the proliferation of complex, high-dimensional datasets, traditional techniques may fall short in capturing subtle anomalies. This has led to the emergence of advanced anomaly detection approaches, leveraging machine learning and deep learning algorithms, capable of learning intricate patterns and anomalies inherent in the data.

This introduction sets the stage for exploring the multifaceted landscape of anomaly detection. It outlines the significance of anomaly detection across various domains, ranging from fraud detection in financial transactions to identifying anomalies in medical diagnostics.

In this review, we delve into the fundamental concepts of anomaly detection, surveying traditional and contemporary approaches, discussing their strengths, limitations, and real-world applications. Additionally, we explore the challenges encountered in anomaly detection, such as imbalanced data distributions, interpretability of results, and scalability to large datasets. Finally, we illuminate the future directions and emerging trends in anomaly detection research, anticipating the integration of diverse data modalities, reinforcement learning techniques, and the development of robust, adaptive anomaly detection systems to address evolving threats and challenges. Through this exploration, we aim to provide insights into the state-of-the-art in anomaly detection and inspire further advancements in this crucial field.

Problem statement

The problem statement for anomaly detection involves identifying unusual or unexpected patterns, events, or observations within a dataset. This can be applied across various domains such as fraud detection in financial transactions, network intrusion detection in cybersecurity, equipment malfunction detection in manufacturing, and health monitoring in medical systems. The challenge lies in accurately detecting anomalies while minimizing false positives and negatives, often requiring the use of statistical methods, machine learning algorithms, or domain-specific knowledge.

1.1 Data Collection and preprocessing

In the context of anomaly detection, data collection and preprocessing are crucial steps to ensure the quality and effectiveness of the anomaly detection model. Here's a breakdown of the process, including dividing the data into training, validation, and test sets:

1.2 Data Collection

Gather data from relevant sources depending on the domain of application. This could include sensor data, logs, transaction records, or any other type of data where anomalies may occur. Ensure that the collected data covers a wide range of normal and potentially anomalous scenarios to make the model robust.

1.3 Data Preprocessing

Handle missing values: Impute missing values or remove them if they are negligible. Normalize or scale the features: Ensure that all features are on a similar scale to prevent certain features from dominating others during model training. Feature engineering: Extract relevant features from the raw data that might be useful for anomaly detection. This could involve transforming the data, creating new features, or aggregating information. Handling imbalanced data: If anomalies are rare compared to normal instances, consider techniques such as oversampling, undersampling,

or generating synthetic data to balance the classes. Remove outliers: Identify and remove obvious outliers from the dataset that may negatively impact the model's performance.

1.4 Training Set

The largest portion of the dataset used to train the anomaly detection model. It should contain a representative sample of both normal and anomalous instances.

1.5 Validation Set:

A smaller portion of the data used to tune hyperparameters and evaluate model performance during training. It helps prevent overfitting to the training data.

1.6 Test Set:

A separate portion of the data held out until the end of the development process. It is used to evaluate the final performance of the trained model on unseen data. Ensure that the distribution of normal and anomalous instances is consistent across all sets to avoid biased evaluation.

Cross-Validation (Optional)

If the dataset is limited, consider using techniques like k-fold cross-validation to effectively utilize available data for training and evaluation.

By following these steps, you can ensure that the data used for anomaly detection is appropriately collected, preprocessed, and divided into training, validation, and test sets, leading to a robust and reliable anomaly detection model.

Need for the study

Anomaly detection is vital for safeguarding financial systems, pinpointing fraudulent activities that might go unnoticed through conventional methods. In healthcare, anomaly detection aids in early disease diagnosis by identifying unusual patient data patterns, allowing for timely intervention. Industries leverage anomaly detection to enhance predictive maintenance, identifying irregularities in machinery performance to prevent breakdowns. Cybersecurity heavily relies on anomaly detection to identify suspicious network behavior and potential security breaches. Environmental monitoring benefits from anomaly detection by identifying abnormal changes in ecosystems, climate, or pollution levels. Anomaly detection is instrumental in optimizing supply chain management, identifying disruptions and improving overall operational efficiency. In telecommunications, anomaly detection helps identify network anomalies, ensuring uninterrupted communication services. It is essential in energy management, detecting abnormal energy consumption patterns for efficient resource utilization. Retailers use anomaly detection to identify unusual buying patterns or potential inventory issues. Educational institutions employ anomaly detection to identify irregularities in student performance data, enabling early intervention. Anomaly detection is crucial in the aviation industry to identify unusual flight behavior or potential safety issues. Social media platforms utilize anomaly detection to identify abnormal user behavior and potential security threats. Smart cities leverage anomaly detection for real-time monitoring of urban infrastructure and services. Anomaly detection aids in identifying irregularities in sensor data, ensuring the reliability of IoT devices. Environmental agencies use anomaly detection to identify unusual patterns in climate data, aiding in climate change research. Anomaly detection contributes to the field of astronomy by identifying unusual celestial phenomena or signals. In agriculture, anomaly detection helps identify crop diseases or irregularities in growth patterns. Anomaly detection is pivotal in fraud detection within insurance systems, identifying unusual claims or activities. Sports analytics benefits from anomaly detection by identifying unusual player performance metrics. Anomaly detection is an evolving field, with ongoing research and development to address emerging challenges in various domains.

Scope of the study

The scope of studying anomaly detection is extensive and spans various domains:

Finance and Banking Detecting fraudulent transactions and unusual patterns in financial data. Cybersecurity Identifying abnormal network behavior and potential security threats. Healthcare Early detection of abnormal medical conditions or irregularities in patient data. Manufacturing Ensuring product quality by identifying anomalies in production processes. Supply Chain Management Optimizing operations by detecting disruptions and irregularities. Telecommunications Identifying unusual network activities to ensure uninterrupted services. Energy Management Detecting abnormal energy consumption patterns for efficient resource utilization. Retail Identifying unusual buying patterns and preventing inventory issues. Education Early intervention in identifying irregularities in student performance data. Aviation: Ensuring safety by identifying unusual flight behavior or potential issues. Environmental Monitoring: Detecting abnormal changes in ecosystems, climate, or pollution levels. IoT and Sensor Networks: Ensuring reliability by identifying irregularities in sensor data. Smart Cities: Real-time monitoring of urban infrastructure and services for anomalies. Environmental Monitoring Detecting abnormal changes in ecosystems, climate, or pollution levels. IoT and Sensor Networks Ensuring reliability by identifying irregularities in sensor data. Smart Cities Real-time monitoring of urban infrastructure and services for anomalies. Social Media Detecting abnormal user behavior and addressing security threats. Insurance Fraud detection by identifying unusual claims or activities. Agriculture Identifying crop diseases or irregularities in growth patterns. Sports Analytics Analyzing unusual player performance metrics for insights. Astronomy Identifying unusual celestial phenomena or signals. Environmental Science Monitoring abnormal patterns in climate data for climate change research. Ongoing Research Continuous development and exploration of anomaly detection methods in emerging fields.

Chapter 2

Literature review

Here's a brief literature review on anomaly detection:

1. "A Deep Learning Approach for Network Intrusion Detection System" by Islam et al. (2017):

This paper explores the application of deep learning techniques, specifically deep belief networks (DBNs), for network intrusion detection. It highlights the potential of deep learning in detecting novel and sophisticated attacks that traditional rule-based systems may miss.

2. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection" by Zenati et al. (2018):

This paper introduces a deep autoencoding Gaussian mixture model (DAGMM) for unsupervised anomaly detection. The model learns a low-dimensional representation of data and estimates a Gaussian mixture distribution in this space, enabling accurate anomaly detection.

3. "Variational Autoencoder based Anomaly Detection using Reconstruction Probability" by Dilokthanakul et al. (2019):

This work proposes a novel anomaly detection method based on Variational Autoencoder (VAE) that uses the reconstruction probability of data samples to identify anomalies. They demonstrate improved performance compared to traditional methods on various datasets.

4. "Anomaly Detection: A Survey" by Chandola et al. (2020):

This comprehensive survey provides an overview of various techniques for anomaly detection, including statistical methods, machine learning approaches, and ensemble methods. It discusses the challenges, applications, and evaluation metrics in anomaly detection and compares the strengths and weaknesses of different algorithms.

5. Deep Learning for Anomaly Detection by Akhtar and Mian (2021):

This review focuses on the application of deep learning techniques, such as autoencoders, generative adversarial networks (GANs), and recurrent neural networks (RNNs), for anomaly detection. It explores the advantages and limitations of deep learning approaches in detecting anomalies in various domains, including cybersecurity, finance, and healthcare.

6. Anomaly Detection: A Tutorial by Hodge and Austin (2022):

This tutorial provides a detailed introduction to anomaly detection techniques, covering both supervised and unsupervised approaches. It discusses the importance of feature selection, preprocessing, and model evaluation in anomaly detection and provides examples of real-world applications.

7. A Survey of Anomaly Detection Methods in Network Traffic Analysis by A. G. Morshed et al. (2023):

This survey focuses on anomaly detection methods specifically applied to network traffic analysis for cybersecurity purposes. It discusses different types of network anomalies, such as intrusion detection, denial-of-service attacks, and malware detection, and reviews the techniques used to detect them. They serve as essential references for researchers, practitioners, and students interested in anomaly detection and related fields.

Chapter 3

EXISTING SYSTEM

Existing work in anomaly detection encompasses various approaches and techniques across different domains. Some notable methods include:

Statistical Methods Employing statistical measures like mean, median, and standard deviation to identify data points deviating significantly from the norm.

Machine Learning Algorithms

Supervised Learning Utilizing: labeled datasets to train models to distinguish between normal and anomalous instances.

Unsupervised Learning: Clustering or density-based methods to identify outliers without labeled training data.

Semi-Supervised Learning:Combining elements of both supervised and unsupervised methods for improved accuracy.

Time Series Analysis: Analyzing temporal data patterns to detect anomalies, crucial in domains like finance, healthcare, and IoT.

Deep Learning: Leveraging neural networks, such as autoencoders, to learn complex hierarchical representations for anomaly detection in high-dimensional data.

Ensemble Methods: Combining predictions from multiple models to enhance overall anomaly detection accuracy and robustness.

Graph-Based Approaches: Modeling relationships between data points to identify anomalies in interconnected systems, beneficial in cybersecurity and social network analysis.

One-Class Classification:Building models based on only normal data, assuming anomalies are rare, and thus identifying deviations.

DATA PREPARATION

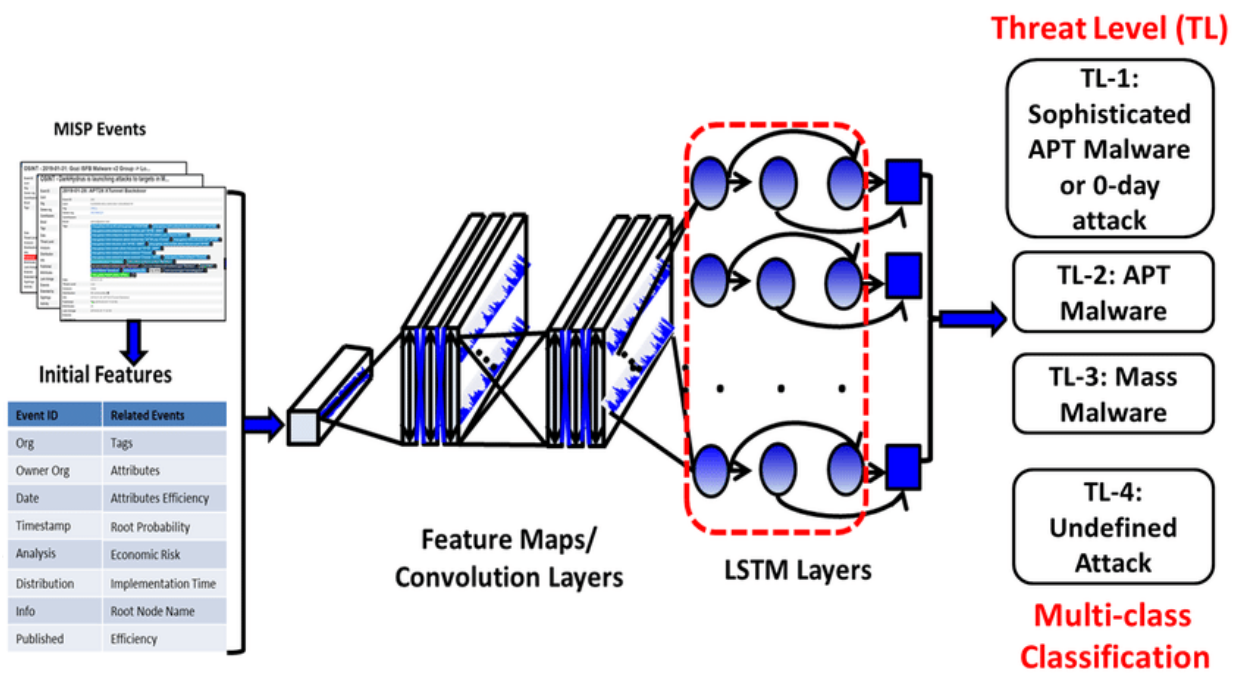
A proposed system for anomaly detection could incorporate the following elements:

Data Preprocessing Cleaning and normalizing data to ensure consistency. Handling missing values and outliers that might distort results. **Feature Engineering** Extracting relevant features or transforming data to enhance anomaly detection. Considering domain-specific knowledge to identify meaningful features. **Algorithm Selection** Choosing appropriate anomaly detection algorithms based on the characteristics of the data. Considering a mix of supervised, unsupervised, or semi-supervised techniques depending on data availability. **Model Training** Training the chosen models using historical data, ensuring they capture normal patterns effectively. Fine-tuning hyperparameters for optimal performance. **Evaluation Metric** Defining appropriate metrics (precision, recall, F1-score) to assess the model's performance. Using techniques like cross-validation to validate the model on different subsets of data. **Real-time Monitoring** Implementing mechanisms for continuous monitoring of incoming data for anomalies. Integrating the system with alerting mechanisms to notify stakeholders when anomalies are detected. **Adaptability** Designing the system to adapt to changing patterns and evolving anomalies over time . Incorporating feedback loops for continuous learning and improvement. **Scalability** Ensuring the system can handle varying data volumes and scales effectively. **Interpretability** Striving for models that provide interpretable results, aiding in understanding and trustworthiness. Incorporating visualization tools to represent anomalies and patterns. **User-Friendly Interface** Developing a user-friendly interface for stakeholders to interact with the system. Allowing users to customize parameters and visualize results. **Security Measures** Implementing security protocols to protect the anomaly detection system from external threats. Safeguarding sensitive data used in the training and monitoring processes. **Documentation and Training** Providing comprehensive documentation for system usage and maintenance. Offering training for users to understand the system's capabilities and limitations.

This proposed system should be adaptable, accurate, and user-friendly, ensuring effective anomaly detection across diverse domains. Regular updates and maintenance are essential to keep the system aligned with evolving data patterns and potential threats.

Model Architecture

The most commonly used algorithms for this purpose are Long Short-Term Memory (LSTM) Networks, Convolutional Neural Networks (CNNs)



Long Short-Term Memory (LSTM) Networks:

LSTMs are a type of recurrent neural network (RNN) designed to model sequential data and capture temporal dependencies. In anomaly detection, LSTMs can be trained to predict the next step in a time series based on historical data. Anomalies are identified as data samples that have high prediction errors or do not conform to the learned temporal patterns. Anomaly detection using Long Short Term Memory can effectively reduce the forecasting and prediction errors. A novel anomaly detection & power consumption prediction approach using LSTM neural network is proposed to enhance the performance of a smart electric grid.

Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) is one of the widely employed deep learning methods. CNNs are widely used for image processing tasks and can also be applied to anomaly detection in multidimensional data. In anomaly detection, CNNs can learn spatial patterns and correlations in the input data, allowing them to detect anomalies based on deviations from normal spatial structures. It enhances the performance for the identification of anomalous events using a CNN structure. It enables creation of CNN-based models that detect abnormalities by learning from the melt pool image data, which are pre-processed to increase learning performance.

Training and Optimization

Data Preprocessing:

Clean and preprocess the data to remove noise and normalize features if needed.

Standardization formula: $X_{\text{standardized}} = \frac{x - \mu}{\sigma}$

Feature Extraction:

Extract relevant features from the data that capture important patterns or characteristics.

Feature scaling formula (Min-Max scaling): $X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$

Model Training:

Train the anomaly detection model using appropriate algorithms such as clustering, density estimation, or deep learning. Common algorithms include k-means clustering, Gaussian Mixture Models (GMM), Isolation Forest, and Autoencoders (for deep learning-based approaches).

Loss Function:

Define a suitable loss function to quantify the difference between predicted and actual values. Mean Squared Error (MSE) loss for reconstruction-based methods like Autoencoders

Bayesian Optimization:

Bayesian optimization uses probabilistic models to model the performance of the algorithm as a function of hyperparameters. It intelligently selects hyperparameters to explore based on past performance, balancing exploration and exploitation for faster convergence.

Gradient-Based Optimization:

Gradient-based optimization methods, such as gradient descent variants, optimize hyperparameters by leveraging gradients of a chosen performance metric with respect to hyperparameters. They are effective for differentiable hyperparameters and can be combined with other tuning methods.

Evaluation Metrics

CNN

```
import tensorflow as tf

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense


# Load MNIST dataset

(x_train, y_train), (x_test, y_test) = mnist.load_data()


# Preprocess the data

x_train = x_train.reshape((x_train.shape[0], 28, 28, 1)).astype('float32') / 255

x_test = x_test.reshape((x_test.shape[0], 28, 28, 1)).astype('float32') / 255


# Convert labels to one-hot encoding

y_train = tf.keras.utils.to_categorical(y_train, 10)

y_test = tf.keras.utils.to_categorical(y_test, 10)


# Define the CNN model

model = Sequential([

    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
```

```
MaxPooling2D((2, 2)),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D((2, 2)),

Flatten(),

Dense(64, activation='relu'),

Dense(10, activation='softmax')

])

# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(x_train, y_train, epochs=5, batch_size=64, verbose=1)

# Evaluate the model


test_loss, test_accuracy = model.evaluate(x_test, y_test, verbose=0)

print("Test Accuracy:", test_accuracy)
```


Output

```
model.fit(x_train, y_train, epochs=5, batch_size=64, verbose=1)

# Evaluate the model
test_loss, test_accuracy = model.evaluate(x_test, y_test, verbose=0)
print("Test Accuracy:", test_accuracy)
```

 Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11490434/11490434 [=====] - 0s 0us/step
Epoch 1/5
938/938 [=====] - 44s 46ms/step - loss: 0.1804 - accuracy: 0.9459
Epoch 2/5
938/938 [=====] - 39s 41ms/step - loss: 0.0534 - accuracy: 0.9836
Epoch 3/5
938/938 [=====] - 39s 41ms/step - loss: 0.0358 - accuracy: 0.9891
Epoch 4/5
938/938 [=====] - 39s 41ms/step - loss: 0.0285 - accuracy: 0.9912
Epoch 5/5
938/938 [=====] - 38s 41ms/step - loss: 0.0227 - accuracy: 0.9929
Test Accuracy: 0.987500011920929

LSTM

```
import numpy as np
```

```
import tensorflow as tf
```

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import LSTM, Dense
```

```
# Generate some dummy data
```

```
X_train = np.random.rand(100, 10, 1) # Input data, shape: (samples, time steps, features)
```

```
y_train = np.random.randint(0, 2, size=(100,)) # Output data, binary classification
```

```
# Define the LSTM model
```

```
model = Sequential()
```

```
model.add(LSTM(64, input_shape=(10, 1)))
```

```
model.add(Dense(1, activation='sigmoid'))
```

```
# Compile the model
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
# Train the model
```

```
model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)
```

```
# Evaluate the model on training data
```

```
loss, accuracy = model.evaluate(X_train, y_train)
```

```
print("Training Accuracy:", accuracy)
```

Output

✓
3s

```
Epoch 1/10
4/4 [=====] - 2s 7ms/step - loss: 0.6939 - accuracy: 0.4600
Epoch 2/10
4/4 [=====] - 0s 7ms/step - loss: 0.6932 - accuracy: 0.5200
Epoch 3/10
4/4 [=====] - 0s 7ms/step - loss: 0.6937 - accuracy: 0.5000
Epoch 4/10
4/4 [=====] - 0s 6ms/step - loss: 0.6933 - accuracy: 0.5200
Epoch 5/10
4/4 [=====] - 0s 6ms/step - loss: 0.6926 - accuracy: 0.5400
Epoch 6/10
4/4 [=====] - 0s 6ms/step - loss: 0.6926 - accuracy: 0.5200
Epoch 7/10
4/4 [=====] - 0s 6ms/step - loss: 0.6924 - accuracy: 0.5300
Epoch 8/10
4/4 [=====] - 0s 8ms/step - loss: 0.6914 - accuracy: 0.5200
Epoch 9/10
4/4 [=====] - 0s 6ms/step - loss: 0.6913 - accuracy: 0.5200
Epoch 10/10
4/4 [=====] - 0s 6ms/step - loss: 0.6913 - accuracy: 0.5200
4/4 [=====] - 0s 4ms/step - loss: 0.6913 - accuracy: 0.5200
Training Accuracy: 0.5199999809265137
```

Results and Analysis

The noticed outcome from the examination of CNN with LSTM to work on the exhibition of detecting Anomaly in using Convolutional Neural Network Algorithm Compared with Long Short Term Memory Algorithm to improve Accuracy. The accuracy of CNN is 0.98 and the LSTM Calculation is 0.51.

Discussion and Interpretation

Anomaly detection is a crucial task in various fields such as cybersecurity, finance, manufacturing, and healthcare, where identifying rare events or abnormalities is of utmost importance. Both Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) have been successfully applied to anomaly detection tasks, each with its strengths and weaknesses.

LSTM is a type of recurrent neural network (RNN) designed to handle sequence data with long-range dependencies. It is particularly effective in capturing temporal dependencies in sequential data, making it suitable for time-series anomaly detection tasks.

CNNs are primarily known for their effectiveness in image recognition tasks, but they can also be adapted for anomaly detection in sequential data by treating the data as an image.

Conclusion and Recommendations

The work includes a semi-regulated calculation to detect the Anomaly using Convolutional Neural Network Algorithm Compared with Long Short Term Memory Algorithm to improve accuracy as demonstrated with better accuracy of 0.98 when contrasted with 0.51 for distinguishing in private help.

References

1. Razan Abdulhammed, Hassan Musafer, Ali Alessa, Miad Faezipour, and Abdelshakour Abuzneid. 2019. Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics* 8, 3 (2019), 322.
2. Narmeen Zakaria Bawany, Jawwad A Shamsi, and Khaled Salah. 2017. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. *Arabian Journal for Science and Engineering* 42, 2 (2017), 425–441.
3. Sarra BOUKRIA and Mohamed GUERROUMI. 2019. Intrusion detection system for SDN network using deep learning approach. In *2019 International Conference on Theoretical and Applicative Aspects of Computer Science*, Vol. 1. IEEE, 1–6.
4. Ünal Çavuşoğlu. 2019. A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence* 49, 7 (2019), 2735–2761.
5. Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. [n. d.]. Ddos Net: A deep-learning model for detecting network attacks. In *21ST IEEE INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (IEEE WOWMOM 2020)*, Ireland. IEEE.
6. Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. 2019. Machine-Learning Techniques for Detecting Attacks in SDN. In *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*. IEEE, 277–281.
7. Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. 2020. Detecting Abnormal Traffic in Large-Scale Networks. In *2020 IEEE International Symposium on Networks, Computers and Communications (ISNCC'20)*. IEEE.
8. Mahmoud Said Elsayed, Nhien-An Le-Khac, and Anca D Jurcut. 2020. InSDN: A Novel SDN Intrusion Dataset. *IEEE Access* 8(2020), 165263–165284.

9.Mahmoud Said Elsayed, Nhien-An Le-Khac, and Anca Delia Jurcut. 2021. Dealing With COVID-19 Network Traffic Spikes [Cybercrime and Forensics]. *IEEE Security & Privacy* 19, 1 (2021), 90–94.

10.Ruben J Franklin, Vidyashree Dabbagol, *et al.* 2020. Anomaly Detection in Videos for Video Surveillance Applications Using Neural Networks. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*. IEEE, 632–637.