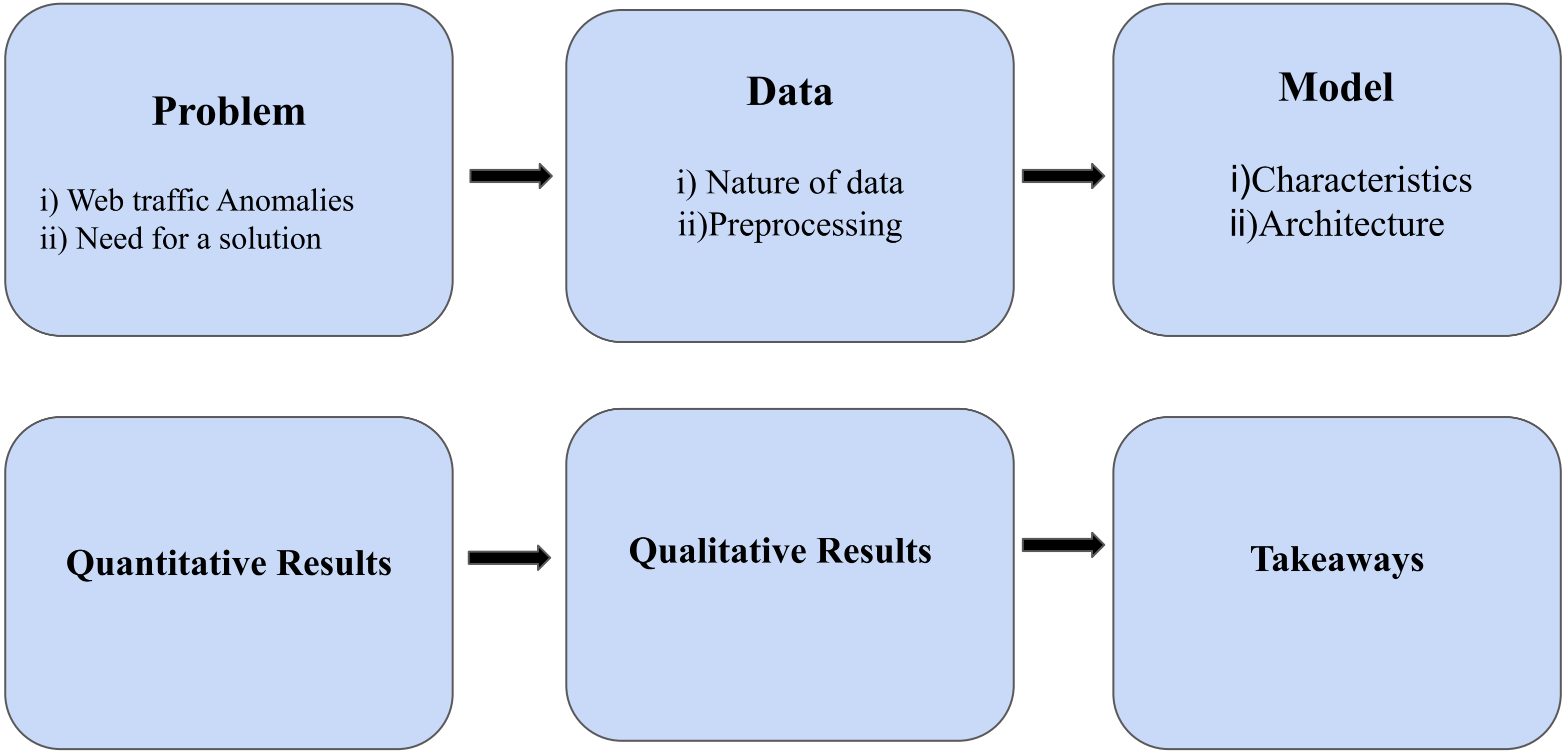
Web traffic Anomaly Detection using C-LSTM Neural Networks

By

Surya V S Bevara	2017A3PS0313P
K Sivakesava	2017A4PS0397P
Sankeerth Nalam	2017A7PS0070P

Outline



Problem

i) Web traffic Anomalies
ii) Need for a solution

Data

i) Nature of data
ii)Preprocessing

Model

i)Characteristics
ii)Architecture

Quantitative Results

Qualitative Results

Takeaways

Issues of Web traffic Anomalies

- The Internet has been a crucial infrastructure in our daily life
- Web traffic refers to the amount of data that is sent and received by people visiting online websites
- Web traffic anomalies represent abnormal changes in time series traffic.
- With the increase in internet services, malicious attacks through networks are gradually becoming more advanced and diversified.
- These attacks can cause serious damage to web service operation, leading to social and economic losses
- We aim to detect these anomalies.

Types of Network Attacks

- **Denial of Service (DoS)**: A malicious attack on a system that causes the system to consume extra resources, which negatively impacts its intended use.

- **Probe:** Collects information about a target network or host and checks which devices are connected to the network.

- **User to Root (U2R):** An attempt to illegally access a managed account to modify, manipulate, or exploit a client’s critical resources

- **Remote to User (R2U):** An attempt to obtain local user access on a target computer and gain permission to send packets over the network.

Various model approaches for web anomaly detection

- As there are serious problems caused by web anomalies, there is a dire need for a effective solution
- Sequence anomaly detection approaches can be divided into three categories:
    - statistical modelling
    - temporal feature modelling
    - spatial feature modelling.

Solution implemented for web anomaly detection

- A C-LSTM neural network that combines a CNN and RNN for automatic feature extraction and detection from web traffic signals is used. Web traffic data is recorded over time and contains specific patterns of spatial and temporal information.

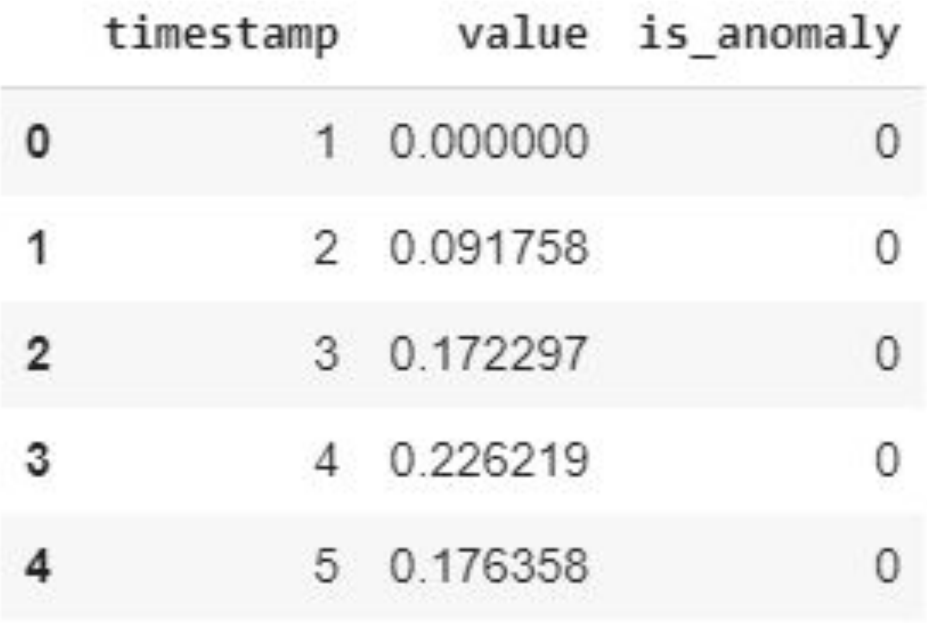
- Spatial features are extracted from time-series data by using a CNN.

- Passing these features through the LSTM helps us to identify how temporal modelling of spatial characteristics in data affects performance.

Nature of the Data

- We utilized the A1 class of the Yahoo Webscope S5 anomaly benchmark dataset, which consists of 67 files, to validate the proposed anomaly detection architecture.
- The files in this class consist of real web traffic data. The data was recorded in 67 files and each file depicted values at various timestamps based on which they were classified into normal/anomaly.
- The collected data is represented by time series of traffic measurement values from actual web services in one hour units.
- Abnormal values were labeled manually and the data has a relatively large variation in traffic compared to other available datasets.

Sample of Data



	timestamp	value	is_anomaly
0	1	0.000000	0
1	2	0.091758	0
2	3	0.172297	0
3	4	0.226219	0
4	5	0.176358	0

Data Preprocessing

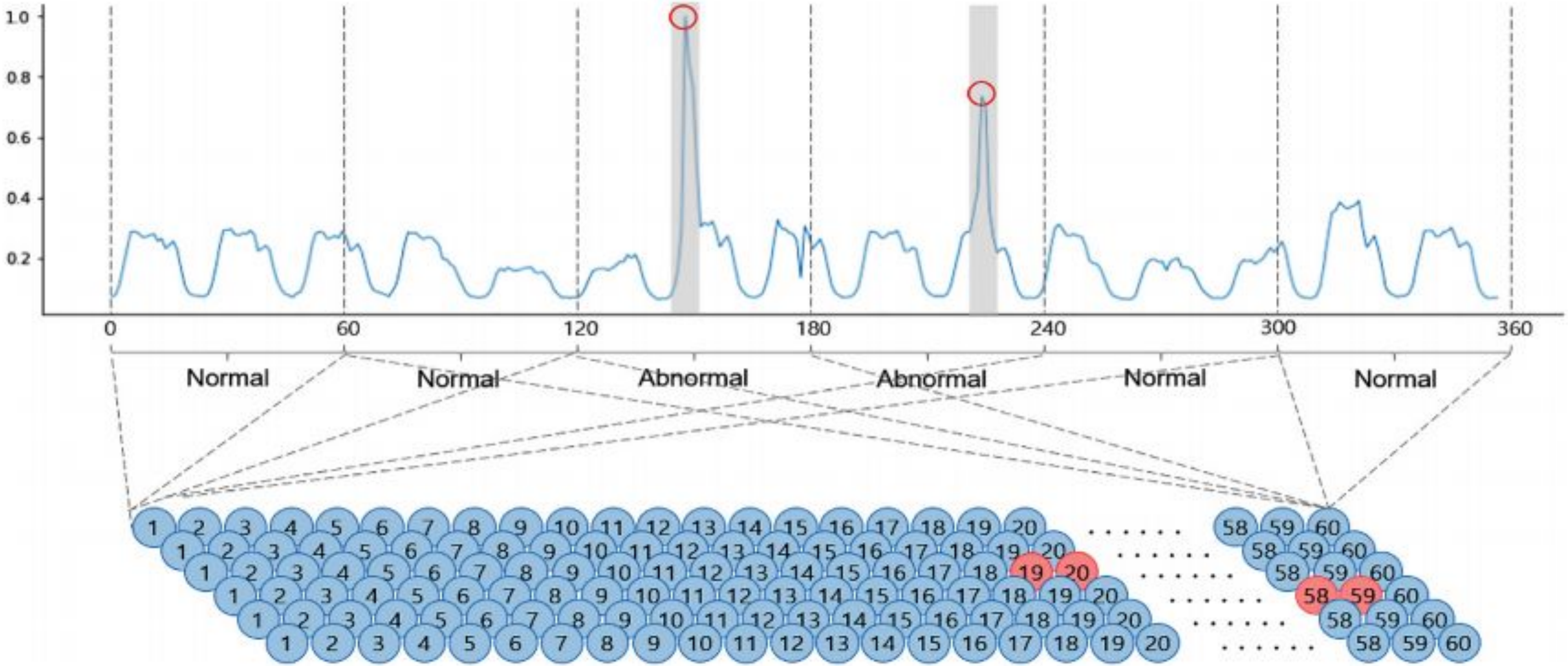
- The Data initially had a few missing values which had to be removed.
- The data has been normalized to maintain uniformity.
- Since the data was in 67 separate files, normalizing them individually and creating windows for each file separately and concatenating them.

Imbalanced Data

- There are a total of 94,866 traffic values in 67 different files, but only 1669 of these values are abnormal.
- The data used has an unusually small ratio of 0.02% abnormal values, meaning the data imbalance is severe .
- Therefore, we applied a sliding window algorithm to solve the data imbalance problem .
- To test the proposed method, 90,910 windows were created by applying the sliding window algorithm and 8470 abnormal windows were labeled.
- The inputs for the C-LSTM are values between 0 and 1, so we had to preprocess the traffic values for anomaly detection. The values were normalized using the following equation.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

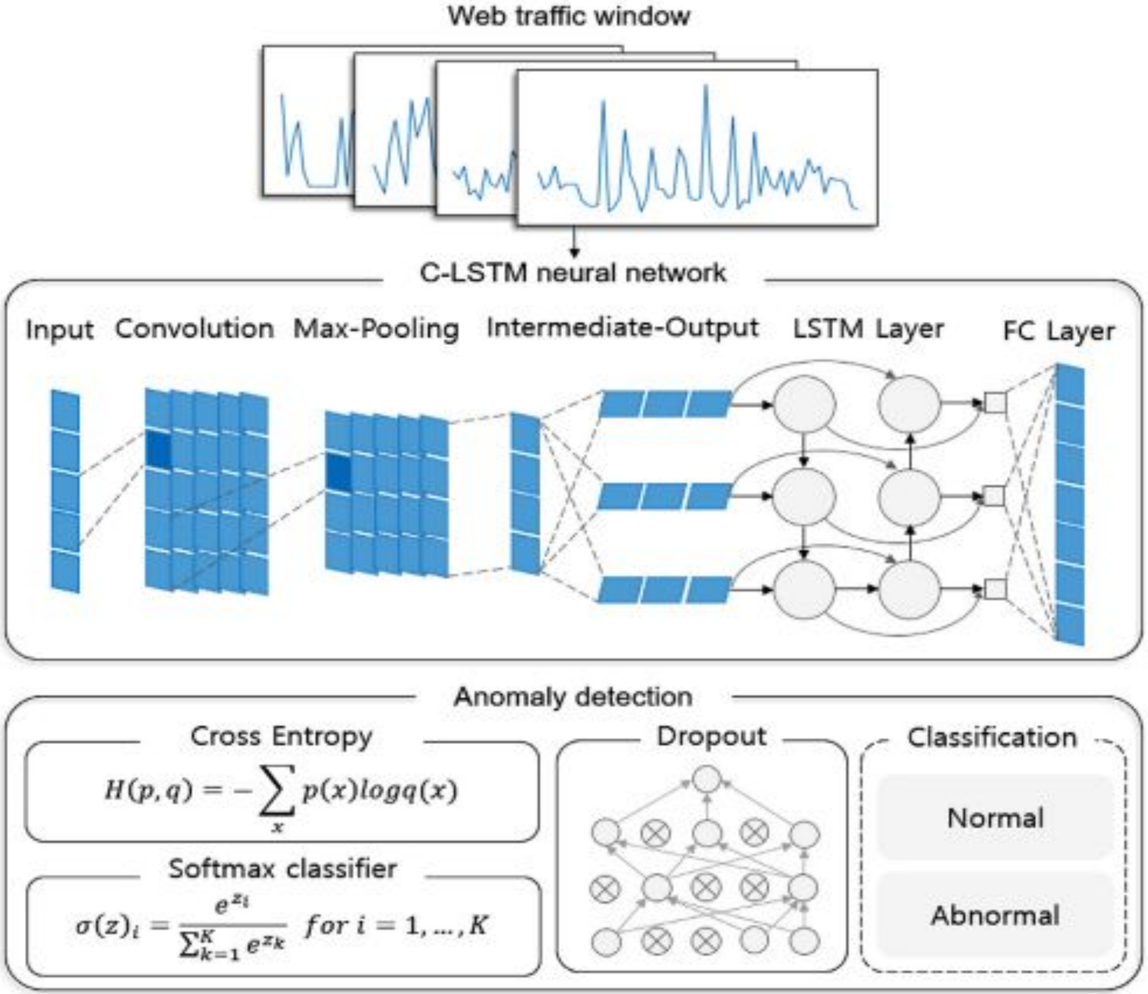
x represents the value of the actual traffic data and x’ represents the normalized value


1.0
0.8
0.6
0.4
0.2
0
60
120
180
240
300
360
Normal
Normal
Abnormal
Abnormal
Normal
Normal
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 . . . . . . 58 59 60

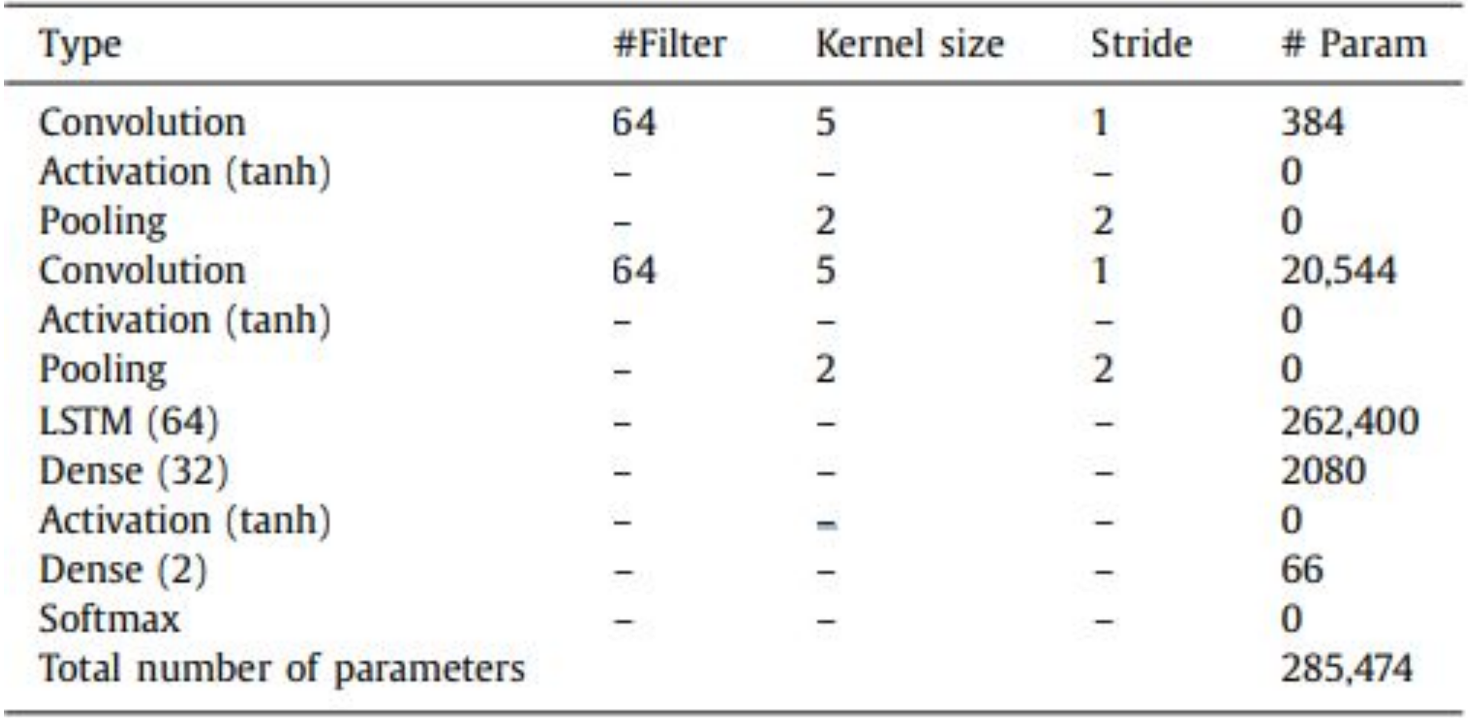
Model characteristics

- The proposed C-LSTM consists of CNN and LSTM layers, and is connected in a linear structure
- The C-LSTM uses preprocessed data as inputs.
- The spatial features in the traffic window are extracted by the convolution and pooling layers.
- The temporal features are then extracted by the LSTM layers.
- A convolution layer is followed by an activation function. This allows the CNN to capture complex features in the input signal

- The input of the C-LSTM is a data vector of length 60 that passes through the LSTM after passing through the convolution and pooling layers.
- We used tanh as an activation function of C-LSTM. Tanh is a function that rescales and shifts the sigmoid function, so that tanh is faster in learning than sigmoid when it is used as an activation function.
- The trained model then performs anomaly detection on the test data using a softmax classifier

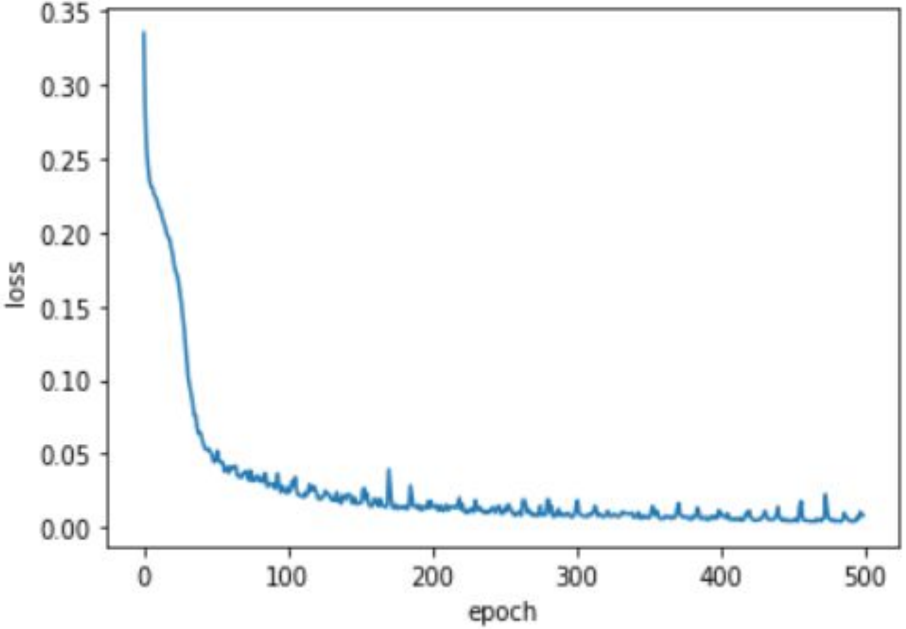

Web traffic window
C-LSTM neural network
Input
Convolution
Max-Pooling
Intermediate-Output
LSTM Layer
FC Layer
Anomaly detection
Cross Entropy
$H(p,q) = -\sum_{x} p(x) log q(x)$
Softmax classifier
$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$ $for\ i = 1, \dots, K$
Dropout
Classification
Normal
Abnormal

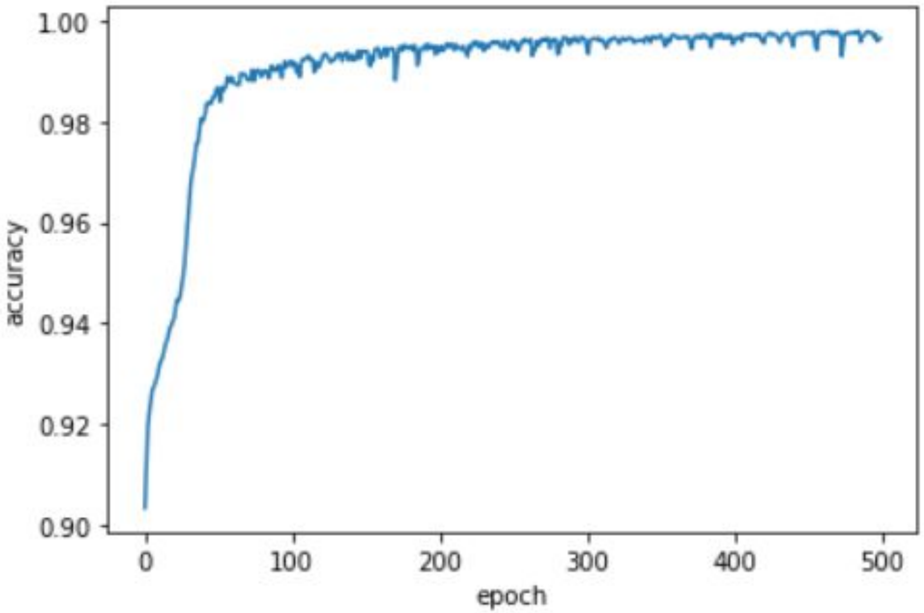
Model Architecture



Type	#Filter	Kernel size	Stride	# Param
Convolution	64	5	1	384
Activation (tanh)	–	–	–	0
Pooling	–	2	2	0
Convolution	64	5	1	20,544
Activation (tanh)	–	–	–	0
Pooling	–	2	2	0
LSTM (64)	–	–	–	262,400
Dense (32)	–	–	–	2080
Activation (tanh)	–	–	–	0
Dense (2)	–	–	–	66
Softmax	–	–	–	0
Total number of parameters				285,474

Results:


0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
loss
0
100
200
300
400
500
epoch


1.00
0.98
0.96
0.94
0.92
0.90
accuracy
0
100
200
300
400
500
epoch

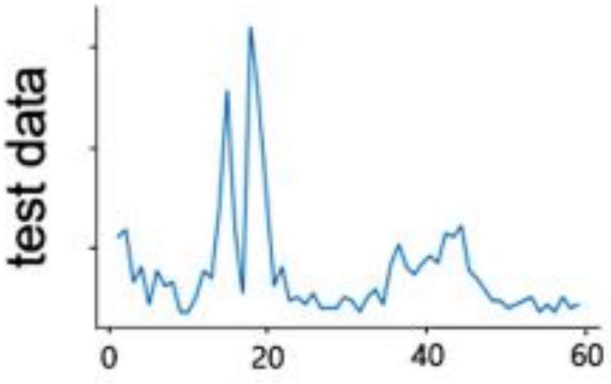
Confusion Matrix

True Predicted	Normal	Abnormal
Normal	22955	1230
Abnormal	704	1729

Performance measures

- **Accuracy**: 92.7%
- **Recall**: 71.1%
- **Precision**: 58.4%
- **F1_score**: 64.2%

Qualitative Results



The model predicts 1 (which means it is an anomaly) which completely makes sense

Comparative Study

1) The accuracy after training with 700 epochs is ~94%.

2) Changing kernel size to 10 improved the model performance by increasing accuracy to 95% and f1-score to 70%.

3) After changing the activation function to relu in every layer, accuracy drastically increased to ~96% and f1-score increased from 64.2% to 73.5%.

Takeaways

- Machine learning algorithms have the ability to learn from data and make predictions based on that data on their own.

- ML model generation and parameter tuning are important but it can’t be the only focus of an ML project. Data gathering and cleansing, integrating ML predictions and feedback loop, and the infrastructure for deploying ML models in production are some of the critical steps that consume more effort.

- Anomaly detection enables us to process data faster,efficiently and securely. Hence, with Artificial Intelligence, businesses can increase effectiveness and safety of their digital operations