Deep Learning Based Disease Outbreak Prediction by Anomaly Detection

Rahul Varma Chintalapati*†, Gaurang Karwande*, Sai Vamsidhar Reddy K, and Siva Krishna Vattikonda
The Bradley Department of Electrical and Computer Engineering, Virginia Tech,
Blacksburg, VA, U.S.A.

***Abstract*—Unsupervised anomaly detection holds a higher significance than its counterpart when the data we have has never occurred before. A prime example is a COVID-19 pandemic. We have never seen such a disease before and thus were not able to predict the consequences of its spreading. After all the planet has experienced, we can safely assume the virus is an outlier among its peers. Our paper is one of the numerous efforts to try and understand the effects and patterns this devastating virus has caused. In this paper, We use various unsupervised machine learning and neural network architectures to detect anomalies in the data which might be an indicator of an outbreak. We use these methods to predict the dates of a potential outbreak by detecting the anomalies in the data. From the Covid -19 Italy dataset, we set aside some of the regions which are peaking at a later date and train our models with the other available regions assuming we have no prior knowledge of these outbreak dates. Experimental results show that each unsupervised learning algorithm used has a specific significance. Some architectures are able to predict the outbreak dates consistently across the regions and others are able to predict the outbreak dates earlier than the others. We use a metric called prediction scores to evaluate the consistency of our models to understand how early our model is predicting the outbreak before the peak occurs. As the most important things are consistency and early detection in predicting the outbreaks, this paper will provide an insight into how various architectures perform and what their results portray.**

***Index Terms*—Anamoly Detection, Deep Learning, Auto-Encoder, Machine learning, COVID-19**

I. Introduction

Anomaly detection is the task of finding outliers in the given data [6]. The general process of anomaly detection systems is to first model the normal behavior of the data and then identify the outliers not conforming to the normal distribution. However, in most real-world scenarios the anomalous data is interspersed with normal data. That is to say, the training data used to model the normal behavior is contaminated with anomalies. In this project, we try different approaches to address this issue.

Today COVID-19 requires no introduction. The whole world is impacted by this devastating pandemic. The disease originated from China in December 2019 and within a span of few months, it evolved into a pandemic. Globally, the outbreak has had many waves. While the first wave caught the whole world unawares, the subsequent waves have been even more devastating. Countries like Brazil and India are severely affected by the second wave. Hence we need to study the pandemic to avoid a repetition of global disruption in the future. A number of studies have conducted to understand the epidemiological dynamics of COVID-19 [1] - [3]. Karadayi et al. presented an outbreak prediction system based on anomaly detection in [4]. They implemented a convolutional LSTM autoencoder network to predict the outbreak dates and our work is based upon their paper. We apply different anomaly detection techniques for the early prediction of outbreak data. Early detection of the outbreak is vital to combating the pandemic. Our specific contributions in the paper include:

- Implementation and comparison of various unsupervised machine learning and deep learning algorithms.
- introducing a performance metric called prediction score to check for the consistency of the algorithms across different regions.

The rest of our paper is arranged as follows: Sec. II discusses the related works, Sec. III discusses about the dataset that was used in the experiments, Sec. IV discusses about the various methods and techniques used in the experiment, Sec. V discusses about the experiments conducted, Sec. VI discusses the results we obtained from the experiments and finally Sec. VII gives a summary and insights on the results, experimentation and future work.

II. Literature Review

Extensive research has been already done in the detection of anomalies in time series data. Anomaly detection is categorized into many ways based on the level of supervision: supervised, semi-supervised and unsupervised. Some of the basic anomaly detection methods used are statistical models, proximity-based models, linear models. In this section, a brief literature review of commonly used anomaly detection methods and some advanced deep learning models for spati temporal data is made.

K-NN anomaly detection method is the simple and most used distance-based anomaly detection method for point anomalies. It is simple, but heavily depends on K value and it is computationally expensive. Density-based anomaly detection models like Local Outlier Factor(LOF)[11][12] and DBSCAN[13] are also used for anomaly detection. There are many other advanced methods derived from mixing the above-mentioned basic methods, LDBSCAN[14] is one such method, it is derived by merging DBSCAN and LOF methods. Cluster-based local outlier factor (CBLOF)[15] is a density-based anomaly detection algorithm, in which the anomaly

* Equal Contribution
† Corresponding author: crahulvarma@vt.edu

score is based on the distance of this instant to the next large cluster. Briant and Kut proposed an ST-Outlier detection algorithm[16], which is derived by modifying a DBSCAN method to detect the spatial neighborhood in the given data. All the above-mentioned algorithms/models follow a similar methodology of detecting the anomalies in the spatial context of data and then using these spatial anomalies to define the temporal anomalies. They do not combine spatial and temporal contexts of data to detect the anomalies. So these methods are not very effective for Spatio-temporal data.

Isolation forests is a method that does not use distance or density-based methods for anomaly detection [9]. It is very effective for time series data having continuous data. This method uses a novel method of isolation trees to detect anomalies. The output of this isolation trees method is the anomaly score of each data point in the data. All the above-discussed methods are unsupervised models. There are some semi-supervised models like one-class SVM(OCSVM)[17]. This method is extensively used for anomaly detection in temporal data, but OCSVM is sensitive to outliers when used for unsupervised data.

Recently, Deep learning models have been extensively used in anomaly detection. Malhotra et al[18]. proposed a model for anomaly detection based on stacked LSTMs. It has a predictive model which is trained on normal time stamps, and the trained model is used to predict the further timestamps, and error is calculated between predicted and original values. This error is used to determine the anomalies in the data. Malhotra et al. also proposed an LSTM encoder-decoder method[19], which reconstructs the normal data, and error is calculated from the reconstructed data. This reconstruction error is used to determine the anomalies. Yang et al.[20] proposed a CNN-LSTM based recurrent autoencoder model for unsupervised extraction of highlights in video data, in this model a trained 3D convolutional network is used for feature extraction for anomaly detection in videos which can be considered as a time series data. Munir et al proposed the DeepAnT model[22], another CNN model where diastance metric is used for determining the anomalies in the time series data. Autoencoders is another deep learning model used for anomaly detection in time-series data. Typically autoencoders are used for dimensionality reduction, but due to the efficiency in encoding for unsupervised data, it is very much effective for anomaly detection in time-series data.

III. Dataset

T The Dataset used by us is an Italian COVID-19 dataset made available by the Department of Civil Protection, Italy. The time-series dataset was created as a national response effort to overcome the COVID-19 emergency. This is a dataset of certain Italian regions that shows the everyday growth of the pandemic. The dataset details epidemiological numbers for all the regions mentioned starting from February 24th until December 6th, 2020.

The Dataset has 8 features: Date, Region code, Region Name, Hospitalized Patients, intensive care patients, current positives, new positive cases, recovered, deaths.Features such as “date, region code, region name, latitude, longitude” are contextual features, and the remaining features are behavioral features. The feature “tests performed” is part of the intervention measures and explicates significant differences between regions as it depends on the respective local authorities. Active testing for COVID-19 and mobilization of resources can affect the dynamics of the pandemic and these are regarded as contextual variables and not regarded as behavioral variables[4]. The above listed features were used and the unnecessary and empty features in the dataset are removed in the preprocessing step of the dataset(”Total number of people tested”,”Country”). From this dataset we have used only the data of the first 150 days- February 24th to July 24th- for training, cross-validation, and as a test set. We have considered the first 150 days as the second peak is about to start roughly after July 24th and making generalizations with multiple peaks would be difficult and does not give accurate results.
The data from Marche is used in cross-validation. As a test set we used the data from the regions - Lazio, Campania, and Sicilia which are from central, southern, and Island parts of Italy respectively.

IV. Methods

Here we briefly introduce some of the widely used machine learning methods for anomaly detection. In the next section, we will explain how each of these is extended to the task of disease outbreak prediction.

A. Autoencoder

An autoencoder is an artificial neural network that non-linearly maps high-dimensional input data into a low-dimensional representation. In contrast, PCA and other matrix methods are used for linear dimensionality reduction. An autoencoder consists of two parts: an encoder that maps the input to the low dimensional latent space and, a decoder that converts this latent space representation back to the input’s feature space. A sample autoencoder is shown in figure 1.

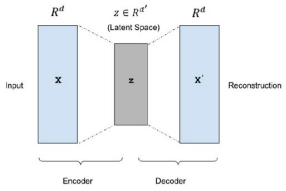

Fig. 1. Autoencoder network

The network takes an input vector $x \in R^d$ and encodes it into a latent representation $x \in R^{d'}$:

$$z = f_\theta(x) = Wx + b \tag{1}$$

The decoder then converts the latent representation back to the input shape $x' \in R^d$:

$$x' = g_{\theta'}(z) = W'x + b \tag{2}$$

The loss function is the measure of dissimilarity between the input and reconstructed output. This is then back-propagated through the network to train the parameters. The basic intuition is that anomalous data will have higher reconstruction loss as compared to the normal data.

B. Isolation forest

Isolation forest is a decision tree based ML algorithm. It builds an ensemble of regressors that recursively separate or isolate anomalies [9]. Other anomaly detection models focus on categorizing the normal data points and then data that cannot be classified as normal is termed as a potential anomaly. But here, it finds the anomalous points first without the need of estimating the normal data distribution. Isolation forest builds an ensemble of random decision trees that split the training data in a recursive way. The intuition behind isolation forest is that an anomalous data point will require fewer splits as compared to normal data. Thus anomalous points are the nodes or leafs with the shortest average path from the root node of overall decision trees. The anomaly score is given as:

$$anomalyscore = 2^{\frac{-E(h(x))}{c(n)}} \tag{3}$$

where $h(x)$ is the number of edges in the tree for a point x and, $c(n)$ is the normalising constant for dataset of size n. $E(h(x))$ is the expected length of the path to reach x overall decision trees. As $E(h(x)) \to 0, anomaly\ score \to 1$.

C. Self-organizing maps(SOM)

Self organizing maps (SOM)[21] is a deep learning architecture that is trained using an unsupervised learning method. It uses competitive learning to adjust weights in the neurons. Each neuron is assigned a weight vector with the same dimensions as the input vector. In the training process, competitive learning follows 3 steps, namely competition, cooperation, and adaptation. In the competition phase, we calculate the distance between each neuron and the input vector and consider the least distant vector as the victor. In the cooperation phase, we select the neighbors of the victor neurons to update those neurons along with the victor neurons in the adaptation phase. This selection is done by considering the time and distance factors. In the adaptation phase, the neighbor neurons' position is shifted closer to the position of the victor neuron, and thus, neurons with similar weights get classified together and anomalies are left out.

D. DB scan

DB scan is an unsupervised machine learning algorithm that clusters data points based on the density metric.In Db scan, we divide data points into clusters and noise data points[13]. Data points that do not belong to any cluster constitute noise. This classification is done based on the minimum number of data points that should be in a cluster and the radius of the cluster.

E. DeepAnT Method

DeepAnT[22] is an abbreviation for Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. As the name suggests, it is a deep learning model, which is modeled using convolutional neural networks. The DeepAnT architecture consists of two parts: Predictor and Anomaly detector. The predictor model is modelled using a 1-D CNN, and the CNN is used for regression instead of classification. The predictor model takes a window and time stamps as input and predicts the next timestamp. Now, in the anomaly detector part, the predicted timestamp is compared with the original timestamp by calculating the distance between these two timestamps. A threshold is set for the difference between the predicted time stamp and the original timestamp, if the distance crosses the threshold, that timestamp is labeled as an anomaly.

V. Experiment

Here we discuss how the techniques discussed above are extended to the task of outbreak prediction.

A. Autoencoder

1) Data Preparation:

For each region, we have a multivariate time series showing the progression of the epidemic. The region data has dimensions of (T, d), where $T = 150$ is the number of timestamps. and the d is the dimension of the observation = 8 vector at each timestamp. We then apply a 7-day window and create temporal subsequences. Thus the regional data is converted into 144 x 7 x 8. For each region, we find 10 nearest regions based upon haversine distance [7]. With these 10 neighboring regions, we create a spatio-temporal dataset that has all the information about the temporal progression of the pandemic across a selected region and its neighbors. Thus corresponding to a single region the Spatio-temporal dataset has dimensions 144 x 7 x 8 x 10. The intuition behind creating such a dataset is that the pandemic situation in a particular city/region is affected by cases within that region as well as by the cases in its closest neighbors as there is always a substantial moment of people between neighboring regions for various essential and non-essential purposes. Thus the overall training dataset has dimensions 1296 x 7 x 8 x 10. The cross-validation dataset has dimensions 144 x 7 x 8 x 10 and the test dataset has dimensions 432 x 7 x 10. The autoencoder model is trained on the training dataset to minimize the mae between actual input and the reconstructed input, the threshold for the reconstruction error is found using the cross-validation dataset and finally, the model performance is evaluated on the test dataset.

2) Model Architecture:

TThe encoder consists of 2 CNN blocks. Each has 2D convolutional layers. A number of feature maps are set to 16 in the first layer and 1 in the second layer. We use a 3 x 3 kernel and set the padding and stride to 1. 2D batch normalization layers are interleaved with the conv2d layers. The activation function is set to ReLU as based upon our

experiments which caused the model to converge faster. The convolutional encoder compresses the spatial information in the data. The decoder consists of 2 LSTM blocks. Each LSTM block has 2 hidden layers. No activation functions were used in the decoder network. Finally, a dense conv1d layer was used to reconstruct the shape of the input. The deep learning framework is optimized with Adam optimizer with a learning rate set to 1e-4. It runs for 100 epochs with batch size set to 16, to avoid any overlap between different regions of the training dataset. The training is regularized by weight decay and early stopping. The complete model was implemented with PyTorch [8].

B. Isolation Forests

Isolation forest implementation in scikit-learn v0.24.2 was used for experiments [10]. This implementation requires the setting of the three parameters: $n_estimators$ is the number of trees that get built, $max\_samples$ is the number of records to draw from training data with replacement, to build each tree and, $contamination\_size$ is the proportion of anomalies in the dataset. Based upon a grid search the $n\_estimators$ was set to 100 and $contamination_size$ was set to 0.1. max_sample was set to $'auto'$. The model was trained on the regions from the training data set, the hyperparameters were chosen using the cross-validation data set. And then prediction for the test regions was obtained using this tuned model. The model outputs an anomaly score for each timestamp and the first anomalous event in the test regions is the predicted outbreak date.

C. Self-organizing maps(SOM)

During training, the model learns the anomalous data points of all the regions in the input data and projects them into a 2D feature map of size 20 x 20. If the data points are normal values, they are clustered together and if the data points are anomalies they are distant from the regular clusters and are shown as white pixels on the feature map [Figure 7]. Once we train the model with Spatio-temporal data, we find out the earliest peak predicted and that corresponding data point is stored as the first anomaly. Then the distance between every data point in each test region and the anomaly data point is calculated and the data point with the minimum Euclidean distance is taken as the predicted first peak of that region. Here we note down and interpret the results We generated a heat map/feature map which gives us the cluster of regular data points and anomalous data points. Then we extract all the possible anomalous data points and check for the first anomaly. Once we obtain the data point we then use this data point and calculate the euclidean distance between all the data points in the test regions and this first anomaly data point. The data point related to the minimum distance that was calculated was considered to be the predicted value of the first anomaly of the test regions. Interestingly, our predicted dates were close to the actual date the Covid-19 cases peaked as we can see in the results. The hyper-parameters used in the experiment are a 20 x 20 feature map, a distance of 1.0, and the number of dimensions was 14.

D. DB Scan

We wanted to have a machine learning architecture as a baseline model to compare our results with and hence chose the DBSCAN clustering algorithm. We ran the model with different values of eps and a minimum number of points and we got optimal results for the hyperparameters with eps as 0.8 and a minimum number of samples as 25. DBSCAN, a spatial clustering algorithm, will cluster highly dense normal points and low-density noise data points. We extract those anomalous data points and choose the first noise point as the first anomalous data point.

E. DeepAnt

DeepAnT architecture is modeled using a 1-D CNN, the CNN has two convolutional layers with 5 input channels, kernel size of 3, which is followed by a max-pooling layer with a kernel size of 2.And this layer is followed by the activation function, the activation function used is ReLU.Finally, all these layers are connected to a fully connected linear layer, which gives an output. The CNN model here is used for regression for forecasting the next timestamp using previous time stamps. The data of five-time stamps is fed into the CNN in parallel, using 5 input channels. Using these 5 timestamps, the CNN predicts the next time stamp. The data is fed into CNN by sliding the input window by one-time stamp at a time. Now, in the anomaly detection part of the architecture, the predicted timestamp values are compared with the original timestamp values using the euclidean distance. A threshold of 0.05 is selected based on the outbreak date of the training data. Now all the anomalous data in the training set is detected and the first timestamp of the anomalies is recorded. The values of all the features used for modeling are taken for this timestamp and these values are compared with the values of the features of all timestamps in the test data by measuring the Euclidean distance between both the timestamps. The timestamp with the least Euclidean distance is then predicted as the outbreak date for that region.

VI. Results

For each of the techniques, the models predict anomalies at a set of dates. The earliest predicted date is then set as the outbreak prediction date. To further validate the predictions and gain intuition about the process we select one of the feature $NewPositiveCases$, and define a new quantity $predictionscore$ which is the ratio between the daily new cases when the peak occurs for that region and the daily new cases at the predicted outbreak date :

$$PredictionScore = \frac{NewPositiveCases\ at\ peak\ of\ pandemic}{NewPositiveCases\ at\ predicted\ outbreak\ date} \quad (4)$$

Figures 3-6 show plot of daily new cases for each of the test regions and the training region 'Piemonte'. The observations at the corresponding predicted outbreak dates are highlighted for each models. The dates and prediction scores are summarized in Table 1 and Table 2 respectively.

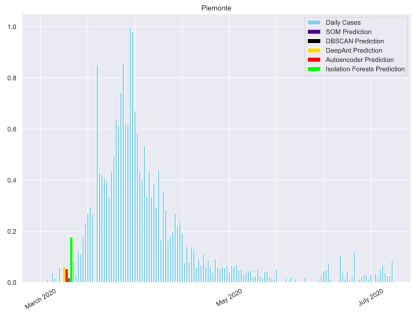

Fig. 2. Piemonte

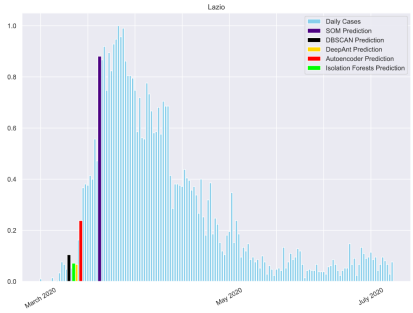

Fig. 3. Lazio

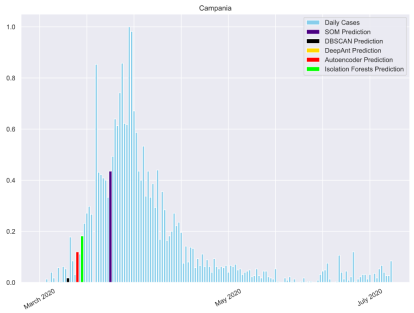

Fig. 4. Campania

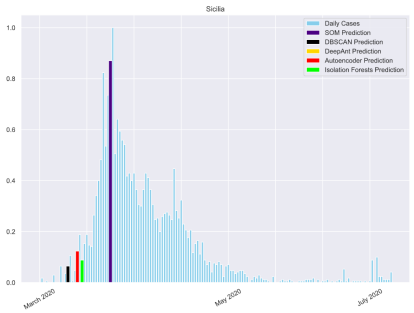

Fig. 5. Sicilia

A. Autoencoder

The final training mae is 3.13 after 86 epochs. We then use the trained model to make predictions over the CV region Marche. Based upon a grid search on these predictions we set the loss threshold to 0.1. Any loss above this threshold indicates an anomalous event occurring on that particular date. The date of the first anomalous event is then the predicted outbreak date. The dates and prediction scores on the test regions are compiled in Table 1 and Table 2 respectively. We see that the prediction scores lie within the same range and are consistent, thus signifying the combination of features at the predicted outbreak dates is comparable between different regions.

B. Isolation Forest

The trained model raises the first alarm signal for Emilia-Rogana and Marche, which are from the training and validation set, on 4th and 3rd March respectively. The predicted dates on the test regions are shown in Table 1 and the prediction scores are shown in Table 2. We see that the prediction scores are quite close for Sicilia and Marche. For Lazio the prediction score is 13.867. While a high prediction score does imply an early detection of the outbreak, this score is not consistent with other two values. A good model should give a high prediction score for all regions and not for only one region. Hence the isolation forest predictions need further analysis.

C. Self Organizing Map

The results are graphically shown in Table 1.When we run our self organizing maps architecture we get a set of anomalous data points depending on how SOM classifies our data.We take those data points and the we select the first anomaly as the tentative date from when the outbreak might begin.From the figures(4,5,6) we can observe that our model is predicting a date at or around the preceding Minima which might imply that it is predicting the steep increase in the number of cases in the upcoming dates and thus we can safely assume the peak is close by.Then we calculated the prediction scores for all the test regions as a sanity check and as a metric to measure the consistency of our model. SOM shows a low prediction score for all the test regions. This means that SOM is predicting the outbreak dates just before the peak occurs.

D. DBSCAN

We got a date of 7 March 2020 for all regions as the predicted date. While this does identify an early detection of the outbreak, further analysis points towards inconsistency in DBSCAN results. The prediction scores are quite dissimilar, hence this raises questions about the DBSCAN predictions and interpretability.

Architecture	Lazio	Campania	Sicilia
SOM	20th March	24th March	25th March
DBScan	7th March	7th March	7th March
DeepAnT	10th March	9th March	11th March
Autoencoder	12th March	8th March	11th March
Islolation Forests	9th March	11th March	13th March

TABLE I
PREDICTED OUTBREAK DATES

Architecture	Lazio	Campania	Sicilia
SOM	1.12	1.928	0.73
DBScan	9.45	37.75	9.909
DeepAnT	14.857	7.947	5.19
Autoencoder	4.16	3.775	5.19
Islolation Forests	13.867	5.59	7.26

TABLE II
PREDICTION SCORES

E. DeepAnT

The DeepAnT architecture predicted 10th march 2020, as the outbreak date for Lazio region, 9th march 2020 for Campania and 11th march 2020 for Sicilia. The ratio of original peak for new positive cases and predicted outbreak for new positive cases are shown in the table below. The DeepAnT model performed well in early prediction of outbreak in regions of Lazio and Campania, but did not do well on Sicilia. In general the model couldn't generalize well on all regions, as the score is not consistent in all regions. Thus, a further analysis is needed for DeepAnT model to use it for detection of outbreaks in pandemics.

VII. CONCLUSION

From all the above experiments we came across quite a few interesting findings. The DB scan clustering algorithm was able to indicate the beginning of the upward trend of the new positive cases. However, it was not accurate enough to predict the early outbreak of Covid-19 cases consistently. Isolation forests algorithm also faces a similar issue of reliability to predict the outbreak of Covid-19 in various regions deemed unsuitable for outbreak prediction analysis. We then moved on experimenting on a bit more complex neural network architectures, namely Self organizing maps, DeepANT architecture, and auto-encoder architecture. SOM predicts the outbreak dates consistently, but it predicts those dates when the region is on the verge of an outbreak. These findings are backed by the prediction scores used as a metric to see how consistently the model predicts dates for different regions. The DeepANT architecture shows promise in predicting the outbreak early. However, it is very fickle in predicting those results consistently. The auto-encoder, however, shows a great promise in the consistency department, but it predictions are not as early as one would like, that is to say the prediction scores can further be increased. From our comparative analysis we see that deep neural network based models like DeepAnt and Autoencoder show promise in outbreak predictions and they can be further improved upon in such way that they are consistent as well as give us earlier predictions.

REFERENCES

[1] A. L. Ziff and R. M. Ziff, "Fractal kinetics of Covid-19 pandemics (with update 3/1/20), doi: 10.1101/2020.02.16. 20023820.

[2] Z. Yang, Z. Zeng, K. Wang, S. S. Wong, W. Liang, M. Zanin, P. Liu, X. Cao, Z. Gao, Z. Mai, and J. Liang, "Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions,doi: 10.21037/jtd.2020.02.64.

[3] L. Zhong, L. Mu, J. Li, J. Wang, Z. Yin, and D. Liu, "Early prediction of the 2019 novel coronavirus outbreak in the mainland China based on simple mathematical model,doi: 10.1109/ACCESS.2020.2979599.

[4] Y. Karadayi, M. N. Aydin and A. S. Öğrencí, "Unsupervised Anomaly Detection in Multivariate Spatio-Temporal Data Using Deep Learning: Early Detection of COVID-19 Outbreak in Italy,"doi: 10.1109/ACCESS.2020.3022366.

[5] The COVID-19 Data Italy Website by Italian Department of Civil Protection. Accessed: May 15, 2020. [Online]. Available: https://github.com/pcm-dpc/COVID-19

[6] Vijay Kotu, Bala Deshpande, Chapter 13 - Anomaly Detection, Editor(s): Vijay Kotu, Bala Deshpande, Data Science (Second Edition), Morgan Kaufmann, 2019, Pages 447-465, ISBN 9780128147610, https://doi.org/10.1016/B978-0-12-814761-0.00013-7.

[7] P. V. Ingole and M. K. Nichat, "Landmark based shortest path detection by using Dijkestra algorithm and Haversine formula," Int. J. Eng. Res. Appl., vol. 3, no. 3, pp. 162165, 2013.

[8] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G. Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[9] Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. "Isolation forest." Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 28252830, Oct. 2011.

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," ACM SIGMOD Rec., vol. 29, no. 2, pp. 93104, Jun. 2000, doi: 10.1145/335191.335388.

[12] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams, doi: 10.1109/CIDM.2007.368917.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd ACM Int. Conf Knowl. Discovery Data Mining, 1996, pp. 226231.

[14] L. Duan, L. Xu, F. Guo, J. Lee, and B. Yan, "A local-density based spatial clustering algorithm with noise, doi: 10.1016/j.is.2006.10.006.

[15] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," doi:10.1016/s0167-8655(03)00003-5.

[16] D. Birant and A. Kut, "Spatio-temporal outlier detection in large databases," J. Comp. Inf. Tech., vol. 14, no. 4, pp. 291297, 2006.

[17] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in Proc. Int. Joint Conf. Neural Netw., vol. 3, 2003, pp. 17411745.

[18] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in Proc. 23rd Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn., 2015, pp. 8994.

[19] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multisensor anomaly detection," presented at the ICML Anomaly Detection Workshop, Jul. 2016.

[20] H. Yang, B.Wang, S. Lin, D.Wipf, M. Guo, and B. Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 46334641.

[22] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, 'DeepAnT: A deep learning approach for unsupervised anomaly detection in time series,doi: 10.1109/ACCESS. 2018.2886457.