Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
17/1 (2024), 1-7. DOI: http://dx.doi.org/10.21609/jiki.v17i1.1172

An Alternative for Kernel SVM when Stacked with a Neural Network

Mgs M Luthfi Ramadhan

Department of Computer Science, Universitas Indonesia, Depok, Indonesia

Email:mgs.m01@ui.ac.id

Abstract

Many studies stack SVM and neural network by utilzing SVM as an output layer of the neural network. However, those studies use kernel before the SVM which is unnecessary. In this study, we proposed an alternative to kernel SVM and proved why kernel is unnecessary when the SVM is stacked on top of neural network. The experiments is done on Dublin City LiDAR data. In this study, we stack PointNet and SVM but instead of using kernel, we simply utilize the last hidden layer of the PointNet. As an alternative to the SVM kernel, this study performs dimension expansion by increasing the number of neurons in the last hidden layer. We proved that expanding the dimension by increasing the number of neurons in the last hidden layer can increase the F-Measure score and it performs better than RBF kernel both in term of F-Measure score and computation time.

Keywords: *point cloud; SVM; segmentation; deep learning; pattern recognition*

1. Introduction

We live in a three-dimensional world, unfortunately, the camera we usually use in general is only able to do a two-dimensional projection that cannot provide information about the position of an object relative to other objects which are actually very important to make machines that perceive the world in three dimensions [1]. However, recently the use of 3D data has become more frequently used mainly point cloud representation. This is caused by increased sensing devices such as Light Detection and Ranging (LiDAR) even on mobile phones with the time-of-flight (TOF) [2] depth camera feature enables easy point cloud acquisition [1]. A point cloud is a collection of points where each point has certain features. Point clouds generally have three features namely x, y, and z coordinates in three-dimensional space, sometimes there are also other features such as color, normal, etc. In the field of geographical information systems (GIS), point cloud data is used to perform the recognition of objects such as buildings, land, plants, and others automatically [3].

Many studies that use point cloud data have been conducted one of them utilizes deep learning. Soilan [4] did point cloud segmentation using PointNet and proved that PointNet which firstly achieved state-of-the-art on indoor datasets can also be used on outdoor datasets. Azady [3] concluded that the color features in point clouds were able to improve PointNet performance in performing segmentation tasks. Gamal [5] conducted building extraction with dynamic graph convolutional neural network (DGCNN) and Euclidean clustering. Unlike Azady, Satria [6] concluded that color features are unable to improve DGCNN's segmentation performance.

All the studies mentioned previously, use softmax (also known as sigmoid on binary classification) on the output layer. The hyperplanes generated by softmax are not optimal in maximizing the margin between classes [7]. This study develops an architecture by combining PointNet and support vector machine (SVM) and will discuss the impact of dimension expansion on segmentation results. We compare dimension expansion by increasing the number of neuron in the last hidden layer against RBF kernel. According to our related works, RBF is the most outstanding kernel for SVM, therefore we use RBF as our baseline method.

The main contributions of this paper lie in the following two aspects:

1) We introduced an alternative for kernel when stacking neural network and SVM.
2) We observed that there are tendencies for the model to have better performance as the dimension is increased.

2. Related Works

The idea of combining neural networks with SVM is by making the neural network as a feature extractor then the resulting feature is forwarded to SVM to be classified. The combination of neural network and SVM has been proven empirically.

Li et al., [8] conducted a classification of atrial fibrillation (AF) recurrence, a condition when abnormal electrical impulses suddenly start firing in the atria. The classification was done using CNN-SVM with RBF kernel. Results show an accuracy score of 96.06% and 93.14% for CNN-SVM and CNN-softmax respectively. Li et al., concluded that CNN-SVM with the RBF kernel was able to do the AF recurrence classification better than CNN-softmax with an improved accuracy score of 2.92%.

Sun et al., [9] conducted a classification of remote sensing images (RIS) on the UC Merced Land Use image dataset using CNN-SVM with RBF kernel. Results show an accuracy score of 96.42% and 92.14% for CNN-SVM and CNN-softmax respectively. Sun et al., concluded that CNN-SVM with the RBF kernel was able to perform RIS classification better than CNN softmax with an improved accuracy score of 4.28%.

Lekha and Suchetha [7] did diabetes classification using CNN-SVM accompanied by experiments on three kernels namely linear, polynomial, and gaussian. Results show the accuracy score of CNN-SVM with the gaussian kernel is 98% while traditional CNN-softmax is 96%. CNN-SVM classification speed is 0.5397 seconds while the classification speed for CNN-softmax is 0.7797 seconds. Lekha and Suchetha concluded that CNN-SVM is better in terms of accuracy and classification speed with an improved accuracy score of 2% and an improved classification speed of 0.24 seconds.

Abien Fred [10] performed intrusion detection on Kyoto University honeypot systems' network traffic dataset using GRU-SVM. The results show an accuracy score of 84.15% and 70.74% for GRU-SVM and GRU-softmax respectively. The GRU-SVM achieved a classification speed of 1.37 minutes while the GRU-softmax classification speed is 1.67 minutes. Abien Fred concluded that GRU SVM is better in terms of accuracy and classification speed with an improved accuracy score of 13.41% and an improved classification speed of 18 seconds.

Zhang et al., [11] performed invasive ductal carcinoma (breast cancer) detection using MSRCNN-SVM. The results show an accuracy of 87.45% and 86.69% for MSRCNN-SVM and MSRCNN-softmax respectively. These results are an average calculation of 5-fold cross-validation. Zhang et al., concluded that MSRCNN-SVM is better than MSRCNN-softmax with improvised accuracy of 0.76%.

3. Methods

This section consists of three parts. The first part explains PointNet, which in this study is used as feature extractor and feature engineering model. The second part explains SVM, which in this study is stacked on top of PointNet as an output layer to perform point-wise classification (point cloud segmentation). The third and final part explains our proposed method in detail.

3.1. PointNet

PointNet was first introduced by Qi et al., [12]. PointNet achieved state-of-the-art results in LiDAR data segmentation problems on the S3DIS dataset [13]. PointNet is capable of processing the raw point cloud along with features at each point without having to do a projection of two dimensions to the point cloud. This model has 2 architectures that are very similar to each other, namely the classification network used for classification problems and the segmentation network used for segmentation problems. In its implementation, PointNet implements convolution, residual connection, and max pooling in its architecture as illustrated in Figure 1 [3].

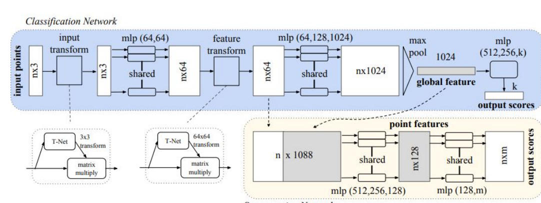

Figure 1. PointNet Architecture [12].

Point cloud classification or segmentation should not be affected by geometry transformations such as rotation, translation, affine transformation, and others. For these problems, PointNet has a T-Net [14] module (transformation network) used to predict the transformation matrix. Then the transformation matrix is used to normalize poses.

In the segmentation architecture, PointNet receives input with the shape of n-point × features and gives output with the shape of n-point × class. It is known as point-wise classification very analogous to pixel-wise classification in image segmentation.

3.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a classification method that is based on a linear function [15]. SVM can do a good classification by utilizing maximum margin. Margins are the distance between hyperplanes with the closest data from each class as illustrated in Figure 2.

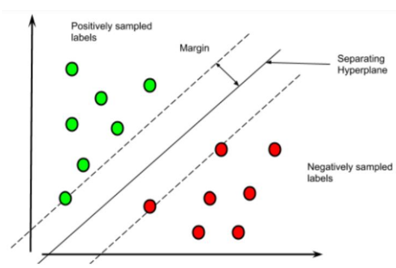

Figure 2. Hyperplane with maximum margin [16].

By maximizing this margin, the hyperplane generated by SVM is the best hyperplane in separating the classes [17]. SVM is a linear classification method that can not classify the data that has non-linearly separable patterns. Kernel functions are the solution to this problem. Kernels will perform non-linear transformations on the data by expanding the dimension into a higher dimension through a non-linear transformation (see Figure 3) so that the data become linearly separable and SVM can classify the data [18].

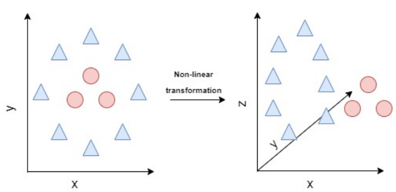

Figure 3. Nonlinear transformation from two dimension into three dimension make the data linearly separable.

3.3. Proposed method

This section consists of three parts. The first part explains about dataset used in this study. The second part explains how we processed the dataset. The third and final part explains how we combine PointNet and SVM to perform point cloud segmentation.

3.3.1. Data accuisition. This study uses a public dataset named Dublin City [19, 20]. This dataset consists of 13 parts as illustrated in Figure 4. Each part has four main labels, namely building, vegetation, ground, and undefined. Of the 13 parts, a hold-out train-test split was carried out by using 8 parts as data training, 1 part as data validation, and 4 remaining parts as testing data. The area inside the green rectangle is training data, the area inside the yellow rectangle is validation data, and the area inside the red rectangle is testing data.

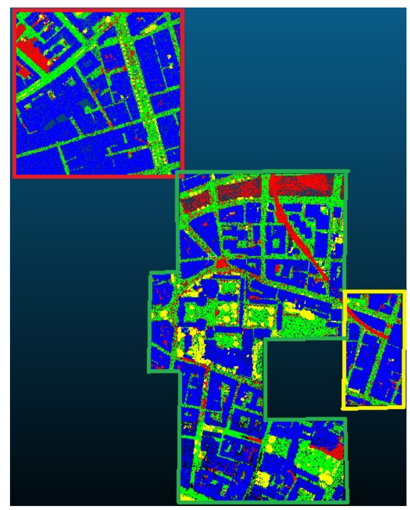

Figure 4. Dublin city dataset and its four main label.

3.3.2. Data preprocessing. We use CloudCompare to perform data preprocessing. Datasets (trains, validation, and test) are sliced with a size of 100 meters × 100 meters, and then each of these slices will be treated as 1 instance or sample in the dataset. Each resulting slice will have a different number of points. To equalize the number of points on each slice, a random sampling of 4096 points was carried out. We decided to use 4096 points as what has been done by [6].

3.3.3. Segmentation. This study proposes an approach by combining PointNet [12] and SVM [21] for point cloud segmentation. The combination is done by implementing SVM as an output layer of PointNet architecture. Both PointNet and SVM are connected end-to-end allowing the gradient to flow from SVM to the first hidden layer of PointNet so that both models can learn simultaneously.

This study does not use the kernel function. Instead, we perform dimension expansion by increasing the number of neurons in the last hidden layer. This idea is based on Cover theory [22] which states

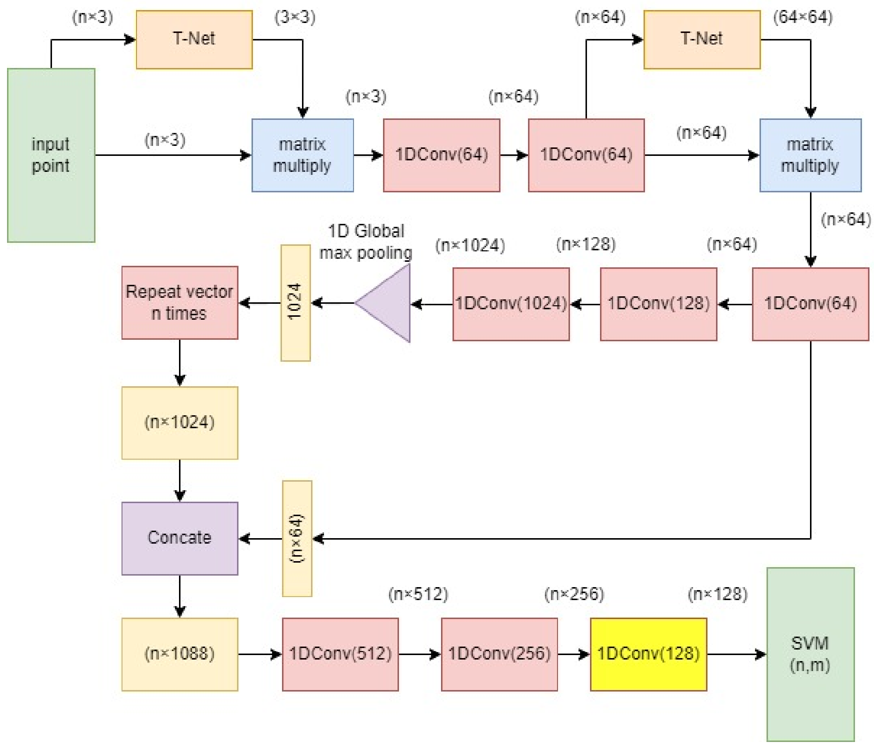

Figure 6. Proposed architecture.

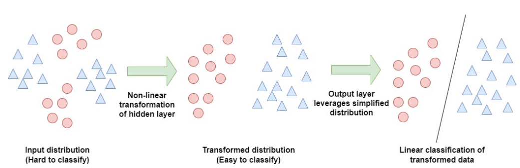

Figure 5. Hidden layer as feature engineering.

that a non-linearly separable pattern when cast into a higher dimension through non-linear transformation will likely be more linearly separable [23]. This same idea is what motivates kernel function for SVM. Goodfellow [15] and Charu [22] explained that the hidden layer itself is already able to represent the data to be linearly separable as illustrated in Figure 5. With that in mind, this study tries to replace the kernel with the approach described earlier. Instead of placing the kernel after the last hidden layer like what many studies have been done before, we simply increase the number of neuron in the last hidden layer.

The overall architecture of our method is illustarted in Figure 6. This model is basically the PointNet which the output layer is replaced by SVM. What we mean by the last hidden layer is indicated by the bright yellow box in Figure 6, this is where the number of neurons is increased as the next layer of this layer is SVM. The shared weight MLP in PointNet (see Figure 1) is theoretically the same as 1D convolution layer with a kernel size of 1 and a stride of 1. For the sake of easier implementation, we illustrated the architecture using 1D convolution.

4. Results and Discussion

Experiments are implemented in Python programming languages and some libraries such as Tensorflow, Pandas, and Numpy. It ran with the following hardware specifications:

- CPU: Intel Xeon
- RAM: 13GB
- GPU: NVIDIA Tesla K80 with 12GB of VRAM

Each model was trained for 25 epochs with the following hyperparameter:

- Optimizer: Adam
- Learning rate: 0.001
- Beta 1: 0.9
- Beta 2: 0.999
- Early stopping patience: 5
- Kernel regularization: L2

Based on the ground truth in Figure 4 which contains very few undefined labels (red). Therefore, F-Measure is used to evaluate the model. In addition to F-Measure, this study also uses segmentation speed which is the time required by the model to perform segmentation in seconds.

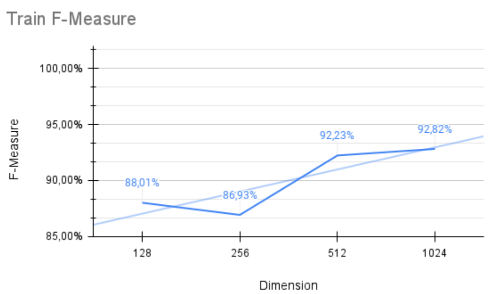

Figure 7. F-Measure score in training set.

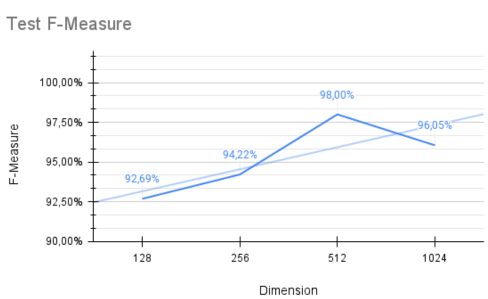

Figure 8. F-Measure score in testing set.

Figure 7 and 8 shows the F-Measure score on training data and testing data respectively. Based on the graph it can be concluded that increasing the dimension give a positive impact on the segmentation performance. The relationship between dimension and F-Measure is analyzed through a linear trend line in the form of $y = mx + b$. A trend line pointing to the upper-right indicates the tendency of the F-Measure increases as the number of neurons in the last hidden layer increases for both SVM.

Figure 9 shows the amount of time each model takes to perform segmentation. Based on the chart, it can also be concluded that the dimensional expansion is directly proportional to the computation time, the higher the dimensions of the data the more time it takes for a model to perform segmentation. Figure 10 shows the total number of epochs to train model. Some of our models are terminated by early stopping. However, there is no meaningful patterns as the trend line is almost flatlined.

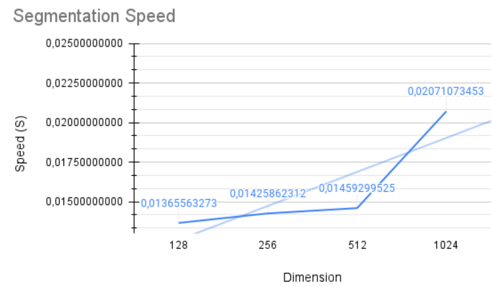

Figure 9. Segmentation speed.

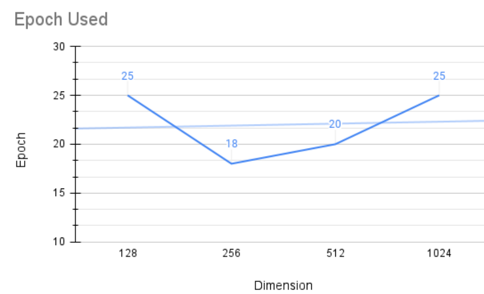

Figure 10. The total number of epochs.

Lastly, the comparison between our approach and RBF kernel is provided in Table 1. Note that, the SVM-128 is actually the original PointNet with SVM as an output layer. The best result is obtained by increasing the number of neuron in the last hidden layer up to 512 neurons. This result outperform RBF kernel both in term of F-Measure and Segmentation speed.

Table 1. Comparison with RBF kernel.

Model	F-Measure	Segmentation Speed (s)
SVM-128	92,69%	**0,0136**
SVM-256	94,22%	0,0142
SVM-512	**98,00%**	0,0145
SVM-1024	96,05%	0,0207
SVM-RBF	91,02%	0.2646

The segmentation speed coorelates with the number of neuron which is no suprise to us since more neuron simply means more computation. We also found out that the best result is obtained by 512 neurons but not by 1024 neurons, yet in the training set 1024 neurons achieved the best result. We believe this is due to the overfitting tendency of a model which fed with high-dimensional data. It is famously known as the curse of dimensionality or the Hughes Phenomenon [24]. Hughes stated that increasing dimension might be beneficial. But when the dimension is too high, a classification model tends to overfit the training set and fails to generalize on the testing set Which is why the 1024 neurons perform worse than 512 neurons.

5. Conclusions

This study examines the impact of dimension expansion by increasing the number of neuron in the last hidden layer as an alternative to kernel for SVM.

Dimensional expansion by increasing the number of neurons in the last hidden layer is proven to be able to increase the F-Measure and it outperforms RBF kernel both in term of F-Measure and Segmentation speed. There is a trade-off between F-Measure and segmentation speed when the dimensions are increased F-Measure will increase, but it worsens the segmentation speed. This matter happens as more and more neurons will cause more computation that occurs in the model.

As far as the author is concerned, this paper is the only study that combines PointNet with SVM, and instead of using kernel, this study simply increase the number of neuron in the last hidden layer.

The most important thing in this study is that we observed it is possible to have a better result by just increasing the number of neurons in the last hidden layer as a replacement of kernel. For now, we hope this study can be useful as a foundation for future studies.

Acknowledgement

The authors wish to thank Muhammad Febrian Rachmadi for his guidance in this study.

References

[1] Manonmani S, Sagar Honnaik, and Shanta Rangaswamy. “Semantic Classification Of Urban Trees Us- ing High Resolution Satellite Imagery”. In: 2018 Inter- national Conference on Electrical, Electronics, Com- munication, Computer, and Optimization Techniques (ICEECCOT). 2018, pp. 161–166. DOI: 10 . 1109 / ICEECCOT43722.2018.9001306.

[2] Horaud, R., Hansard, M., Evangelidis, G., and Ménier, C. (2016). An overview of depth cameras and range scanners based on time-of-flight technologies. Machine Vision and Applications, 27(7), 1005–1020. https://doi.org/10.1007/s00138-016-0784-4.

[3] Bayu, A., Wibisono, A., Wisesa, H. A., Intizhami, N. S., Jatmiko, W., and Gamal, A. (2019). Semantic Segmentation of Lidar Point Cloud in Rural Area. 2019 IEEE International Conference on Communication, Networks and Satellite, Comnetsat 2019 - Proceedings, 73–78. https://doi.org/10.1109/COMNETSAT.2019.8844074.

[4] Soilán, M., Lindenbergh, R., Riveiro, B., and Sánchez-Rodríguez, A. (2019). POINTNET for the AUTOMATIC CLASSIFICATION of AERIAL POINT CLOUDS. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4(2/W5), 445–452. https://doi.org/10.5194/isprs-annalsIV-2-W5-445-2.

[5] Gamal, A., Wibisono, A., Wicaksono, S. B., Abyan, M. A., Hamid, N., Wisesa, H. A., . . . Ardhianto, R. (2020). Automatic LIDAR building segmentation based on DGCNN and euclidean clustering. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00374-x.

[6] Wicaksono, S. B., Wibisono, A., Jatmiko, W., Gamal, A., and Wisesa, H. A. (2019). Semantic Segmentation on LiDAR Point Cloud in Urban Area using Deep Learning. 2019 International Workshop on Big Data and Information Security, IWBIS 2019, (2018), 63–66. https://doi.org/10.1109/IWBIS.2019.8935882.

[7] Lekha, S., and Suchetha, M. (2018). A novel 1-D convolution neural network with SVM architecture for real-time detection applications. IEEE Sensors Journal, 18(2), 724–731. https://doi.org/10.1109/JSEN.2017.2780178.

[8] Li, Z., Feng, X., Wu, Z., Yang, C., Bai, B., and Yang, Q. (2019). Classification of atrial fibrillation recurrence based on a convolution neural network with SVM architecture. IEEE Access, 7, 77849–77856. https://doi.org/10.1109/ACCESS.2019.2920900.

[9] Sun, X., Liu, L., Li, C., Yin, J., Zhao, J., and Si, W. (2019). Classification for Remote Sensing Data with Improved CNN-SVM Method. IEEE Access, 7, 164507–164516. https://doi.org/10.1109/ACCESS.2019.2952946.

[10] Abien Fred, M. A. (2017). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. ArXiv, 2006, 26–30.

[11] Zhang, J., Guo, X., Wang, B., and Cui, W. (2021). Automatic Detection of Invasive Ductal Carcinoma Based on The Fusion of Multi-Scale Residual Convolutional Neural Network and SVM. IEEE Access, 9. https://doi.org/10.1109/ACCESS.2021.3063803.

[12] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). PointNet: Deep learning on

point sets for 3D classification and segmentation. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January, 77–85. https://doi.org/10.1109/CVPR.2017.16.

[13] Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 1534–1543. https://doi.org/10.1109/CVPR.2016.170.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In NIPS 2015

[15] Goodfellow, I., Bengio, Y., and Courville, A. 2016. Deep Learning. MIT Press, London, England.

[16] Chakravarti, M., & Kothari, T. (2015). A Comprehensive Study On The Applications Of Machine Learning For Diagnosis Of Cancer. (May 2015), 0– 19. Retrieved from http://arxiv.org/abs/1505.01345.

[17] Darmatasia, & Fanany, M. I. (2017). Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM). 2017 5th International Conference on Information and Communication Technology, ICoIC7 2017. https://doi.org/10.1109/ICoICT.2017.8074699.

[18] Mohri, M., Rostamizadeh, Y., & Talwalkar, A. 2018. Foundations of Machine Learning. MIT Press, London, England.

[19] Iman Zolanvari, S. M., Ruano, S., Rana, A., Cummins, A., Da Silva, R. E., Rahbar, M., & Smolic, A. (2019). DublinCity: Annotated LiDAR point cloud and its applications. ArXiv, 1–13.

[20] Laefer, D. F., Abuwarda, S., Vu, V. A., Truong-Hong, L., & Gharibi, H. (2017). 2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record. (June), 0–8. https://doi.org/doi:10.17609/N8MQ0N.

[21] Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh (1992).

[22] Charu C. Aggarwal. 2018. Neural Networks and Deep Learning. Springer International Publishing AG, Gewerbestrasse, Switzerland.

[23] T. M. Cover, ”Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition,” in IEEE Transactions on Electronic Computers, vol. EC-14, no. 3, pp. 326-334, June 1965, doi: 10.1109/PGEC.1965.264137.

[24] G. Hughes. 1968. On the mean accuracy of statistical pattern recognizers. IEEE Transactions on Information Theory. https://doi.org/10.1109/TIT.1968.1054102.