https://doi.org/10.31449/inf.v45i4.3747 Informatica **45** (2021) 643–652 **643**

Impact of Gaussian Noise for Optimized Support Vector Machine Algorithm Applied to Medicare Payment on Raspberry Pi

Shrirang Ambaji Kulkarni, Varadraj Gurpur, Christian King and Andriy Koval
School of Global Health Management and Informatics, University of Central Florida, 32816, Orlando, Florida, USA
E-mail: sakulkarni@ucf.edu, varadraj.gurupur@ucf.edu, christian.king@ucf.edu, Andriy.v.koval@gmail.com

Keywords: medicare analysis, internet of things, data, statistical feature optimization techniques, support vector machine, pipelined models

Received: September 11, 2021

A relatively large dataset coupled with efficient but computationally slow machine learning algorithm poses a great deal of challenge for Internet of Things (IoT). On the contrary, Deep Learning Neural Networks (DLANNs) are known for good performances in terms of accuracy, but by nature are computationally intensive. Based on this argument, the purpose of this article is to apply a pipelined Support Vector Machine (SVM)) learning algorithm for benchmarking public health data using Internet of Things (IoT). Support Vector Machine (SVM) a very good performing machine learning algorithm but has constraints in terms of huge training time and its performance is also susceptible to noise. The applied software pipelined architecture to SVM was to minimize its computational time under a resource constrained device like raspberry pi. It was tested with a medicare dataset with Gaussian noise to assess the impact of noise. The classification results of Total Medicare Standardized Payment Amount obtained indicated that the proposed pipelined SVM model was optimal in performance compared to DLANN model by 79.74% in terms of computational time. Also the performance of SVM in terms of area under curve (AUC) was better compared to other models and outscored Logistic Regression by 7.2%, and DLANN model by 22.65%.

Povzetek: Analiziran je vpliv Gaussovega šuma na SCM metodo za plačevanje medijskih storitev.

1 Introduction

Allhoffa and Henschke indicate that [1] Internet of Things (IoT) will become one of the greatest technologies that will revolutionize information capabilities and will have tremendous impact on the society at large. It is to be noted that IoT has limitations in terms of processing, memory and secondary storage capacities as compared to laptops, workstations and servers. Haller et al. [2][3] define IoT as "a world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business process." On the other hand, Gokhale et al., [4] define IoT simply as a "network of physical objects." Here they indicate that generally speaking devices, vehicles, buildings and other forms of hardware and their embedded software can be conceived as physical objects. IoT has be of special importance to the world of healthcare where organizations pertaining to the healthcare ecosystem are working towards reduction of costs and improving productivity. IoT is especially useful in decision support, transmitting information, and device control. Much of this pertains to the field of healthcare informatics. Healthcare informatics is defined by Wan and Gurupur [5] as "a transdisciplinary study of the data flow and processing into more abstract forms such as information, knowledge, and wisdom along with the associated systems needed to synthesize or develop decision support systems for the purpose of helping the healthcare management processes achieve better outcomes in healthcare delivery." The processes involved in synthesizing and developing decision support systems from knowledge and information requires innovative computational solutions and bolsters the need to advance data science especially pertaining to machine learning. Machine learning can be effectively performed in a suitable computational environment.

It is to be noted that edge computing or fog computing is becoming popular day by day as advanced biomedical devices are involved in collecting patient medical data thereby further improving processes associated with healthcare delivery. The advantages in terms of reduced latency between users, edge infrastructure and cloud are evident as described by Shukla et al., [6]. The central storage and sophisticated processing facilities provided by cloud facilities at time may suffer from network latency issues for real-time applications and may act as a single point of failure. It is to be noted that Machine Learning (ML) algorithms are being applied in plethora of applications in relation to the context discussed.

In the work delineated in this article the investigators explore Raspberry Pi as an edge computing device for benchmarking a popular ML algorithm Support Vector Machine (SVM). The SVM is defined by Noble [7], as "a

computer algorithm that learns by example to assign labels to objects." As explained by Noble [7] SVM is a key algorithm that can be effectively used to identify patterns that can be used to train and label data for the purpose of classification. Here the classifiers performance is measured using the concept of Area under the Curve (AUC) as explained by Bradley [8]. This attribute brings about a key desired characteristic for analysing healthcare data. In the recent past many investigators have used the combination of Raspberry Pi and SVM to identify noise and patterns.

For experimentation and demonstration the investigators have used health care data with 40,662 rows and 28 variables, logistic regression algorithm for computational time and Deep Learning Neural Network (DLANN) for testing the accuracy of the classification results of Total Medicare Standardized Payment Amount. The reason for choosing SVM is its ability to produce results at higher level of accuracy; however, SVM tends to be constrained by high computational time and memory complexities for larger size training data [9]. This problem is compounded by the constrained computational resources of a Raspberry Pi and the presence of noisy data. The solution explored is an application of the pipeline architecture for SVM and its performances against the benchmarks set by of logistic regression and deep learning neural network on the same dataset.

The specific research objectives of the analysis are as follows:

- To analyse the performance of SVM with other benchmarks such as Deep Learning Machine Algorithm, and Logistic Regression in terms of accuracy and computational time under optimized and selected variable dataset for a resource constrained environment of Raspberry Pi and
- To implement a pipelined architecture model for SVM with feature selection and ascertain the consistency of performance in terms of metrics and robustness by evaluating the performances on a Gaussian Noise based dataset.

The presentation of a pipelined architecture is to contribute to the science of applying SVM to Medicare and Medicaid type datasets. Here the investigators are mindful of the fact that different datasets of different sizes and complexities require different approaches for analysis in terms of machine learning. More importantly it is important to state that the key targeted contribution of the experimentation explained in this article is to provide a computational method that can be effectively used in analysing healthcare data.

2 Related work

SVM suffer from high time required for training datasets [9][10]and memory complexities issues. These problems are compounded for large datasets and for noisy data were SVM had disadvantages in terms of performance, SVM was applied by Cheng-Lung Huang [11] for credit scoring. They proposed a SVM with Genetic Algorithms (SVM-GA). One of the drawbacks which they observed that SVM-GA took large training times and proposed SVM-GA to be suitable for parallel architectures. Yazici et.al [12], in their work observed the performances of machine learning algorithms on raspberry pi as a part of their study on edge computing paradigm. Some of their results proved that SVM algorithm was slightly faster in inference and also efficient in power consumption. The above work's motivated us to reduce SVM's computational time by integrating it with a pipeline architecture model for working on moderately large datasets for a resource constrained environment like raspberry pi.

Nguyen and Torre [13] in their work discussed that feature selection aided Support Vector Machines towards generalization and computational efficiency. The authors proposed a convex energy-based framework towards feature selection and parameter selection. Experiments on seven different datasets and with feature selection helped them to retain the desired performances. Sanz et.al, [14] discussed in their work that predictor models with most relevant variables was one of the important criteria for biomedical research. They proposed the extension of Recursive Feature Elimination (RFE) based on non-linear SVM kernels. The proposed methods when applied on 3 different datasets performed better as compared to classical RFE.

Logistic regression a supervised learning is one of the popular models applied for classifying medical healthcare data. Logistic regression usually works on large sample size and thus the motivation to apply the same to our 2014 Medicare Provider Utilization and Payment Data [15]. Zardo and Collien [16] successfully used logistic regression to successfully identify critical predictor variables in public health policy research in Australia. Incidentally, Sheets et.al, [17] demonstrated the use of logistic regression in identifying attributes associated with high utilization of Medicare payments, thereby creating a burden on US taxpayer dollars. This research is focused on chronic patients and managed care and proactively identify high risk patients to reduce the cost of healthcare. Thus the present study would like to extend logistic regression to resource constrained environment of raspberry pi.

Deep Learning Artificial Neural Networks (DLANN) are more specialized forms of artificial neural networks and can also learn on their own and handle huge datasets to provide superior classification accuracy, but they also need huge computational resources. Sakr et.al, [18] in their work applied Convolutional Neural Networks (CNN) and SVM for automation of sorting waste on raspberry pi 3.SVM appeared to have higher classification accuracy as compared to CNN by outscoring CNN by 11.8%. Ravi et.al, [19] also studied the impact of Deep Learning algorithms on Health Informatics. They summarized that most of the deep learning algorithms were applied to balanced or synthetic datasets. Also, deep learning algorithms required large amounts of training data.

Thus, with algorithms like logistic regression, deep learning the investigators would like to benchmark the classification accuracy and related performances of support vector machine on a pipeline architecture on a resource constrained device like raspberry pi which holds lot of promise for edge devices. This analysis was carried

on a dataset of 40,662 records [15]. Gangsar and Tiwari [20] studied the impact of noise for fault diagnosis of electric machines. They found for perfect original signal SVM predicted with greater accuracy for all speeds. However, when white Gaussian noise was applied to the raw signal, the overall prediction accuracy fell by 10%. They considered 2% external noise for their study. Pei et.al, [21] in their studies considered the impact of images with white Gaussian noise and their performance effects on convolutional neural networks (CNNs). As the percentage of noise addition increased, the accuracy started to decrease. Wu and Zhu[22] analysed real world data in terms of noise handling features of data mining algorithms. They said error-aware data mining algorithms improved the data mining results. Last but not the least, in their work Zualkernan et.al., [23] considered the application of remote cameras for monitoring animals. They considered an IoT based system whereby images captured on a camera are processed on the edge using Raspberry Pi and the accuracy results are moved to the cloud database system. To summarize application of SVM and other methods related to data science has immense potential that needs to be further explored and the experimentation presented in this article is a step taken in that direction.

3 Method and experiments

The block architecture of the experimental setup is as illustrated in Figure.1

The experiments were executed once the platform was laid, this included implementing the pipelined model for SVM, installing tensor flow for Deep Learning algorithms and a computational time model on a resource constrained environment of Raspberry Pi.

3.1 Statistical optimization and performance

The dataset used for experimentation is a medical healthcare data that contains records for physical therapy patients and amounts paid to the physical therapists in each case Gurupur et al., [15]. It becomes imperative to consider feature section techniques for dataset pruning as

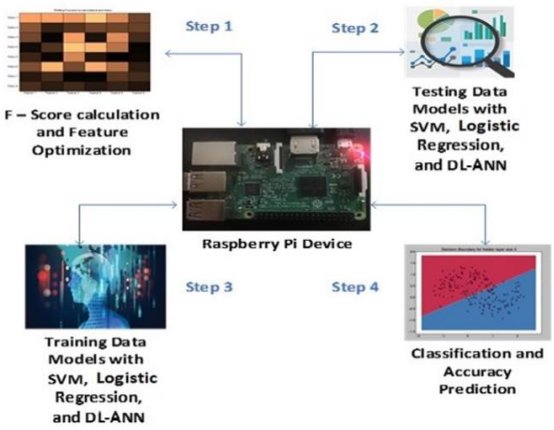

Figure 1: Block architecture of the experimental setup.

an optimization technique for resource constrained environment. Hardware Platform used for the experiment was Raspberry Pi B; Quad Core ARM Cortex A53 CPU 1.2GHz 64bit CPU with 1 GB RAM. From a software perspective, a python program was written using numpy, pandas and scikit-learn [24] along with keras and tensorflow; to apply logistic regression, SVM and DLANN for all variables in order to model them as a classification problem under supervised learning. This software platform was also used to execute metrics like K-Fold Cross Validation, Confusion Matrix and Area Under Curve (AUC).

The reason for applying statistical techniques for the dataset as follows:

a) To optimize the data features so that it helps the machine learning algorithm to classify with a lesser number of variables.
b) To identify outlier's and remove those from the dataset so that we have statistically a more normalized dataset.

Feature selection is an important step in the application of machine learning to achieve at times better performance from the models in terms of computational execution speed. The presence of irrelevant features may negatively affect this application. This creates the need for developing parsimonious models. The advantages could be minimizing the impact of overfitting, accurate results and reduce timing. Therefore, feature selection was the first step in the process. This was implemented using Python scikit-learn library [24] that provides a class called SelectKBest and to this the investigators further utilized the f_classif score function. Finally, SelectKBest retains the first K features of the input dataset X minus the target variable. In our case the value of k was 10. Using this process the investigators listed the features with top 10 F_Score in Table 1.

This was followed by the statistical determination of the presence of outliers [25]. As defined by Zhao [26] an "outlier is considered as a data point which is far from other observations." Here the investigators believe that the presence of outliers may have an impact on the final results of machine learning models. With this in mind, the investigators applied Interquartile range (IQR) to detect the presence of the outliers. Technically, as applied in [27] the IQR is measured as the difference between the third Quartile and the first Quartile i.e. IQR = IQ3-IQ1. After applying the operation to remove outliers from the dataset the investigators removed 6,579 entries.The skewness of the dataset was measured. Skewness as indicated by [27] attempts to indicate the normal distribution of the values. Finding outliers and removing them from the dataset is one of the ways of handling skewness, this process was outlined by [29]. Thus, we measure skewness of the selected features before and after removing outliers from our dataset (Table 2).

It can be observed in Table 2 that after removing outliers the skewness of the selected features has reduced. The analysis of binary classification for selected variables for logistic regression, SVM and DLANN is as illustrated in Figure 2.

Metrics applied were K-Fold validation test, confusion matrix metrics and Area Under Curve (AUC). Cross validation is used to gauge the effectiveness of the model. It involves using a sample of the dataset for testing and training the model on the remaining part of the dataset [30]. The value of k determines the number of groups that a data can be split into. In our case we have set the value of k to 10; therefore, the name 10-fold cross-validation.

Additionally, investigators have used a confusion matrix also termed as an error matrix to analyse the performance of a machine learning algorithm in a matrix format [31]. It is as shown in Table 3.

In the confusion matrix, TP stands for true positive, TN stands for true negative, FN stands for false negative and FP stands for false positive. The assumptions made are S_{TP} denotes the Samples of True Positive, S_{TN} are the samples which denote True Negatives, S_{FP} denotes the Samples for False positive and S_{FN} gives the samples for False Negatives.

Feature variable names	**F_Score**
Number of Services	22369.69
Total Medicare Standardized Payment Amount	22184.17
Total Medicare Allowed Amount	22119.67
Total Submitted Charge Amount	19193.84
proxy for # of new patients	19177.12
Number of Medicare Beneficiaries	18581.63
Average Medicare Standardized Amount per Beneficiary	7535.67
Number of HCPCS	6275.17
Physical therapy services that involve Physical Agents	1998.79
Physical therapy services that involve Therapeutic Practice	1998.79

Table 1: Feature selection based on F-Score.

Feature variable names	**With Outliers Skewness**	**Without Outliers Skewness**
Number of HCPCS	0.59	0.26
Number of Medicare Beneficiaries	2.70	0.98
Average Medicare Standardized Amount per Beneficiary	2.05	0.66
Physical therapy services that involve Physical Agents	1.53	1.17
Physical therapy services that involve Therapeutic Practice	-1.53	-1.17
proxy for # of new patients	2.87	0.78
Number of Services	3.96	1.06
Total Submitted Charge Amount	3.97	1.07
Total Medicare Allowed Amount	4.15	1.01
Total Medicare Standardized Payment Amount	4.55	1.05

Table 2: Measuring skewness with and without outliers.

```
Input: Medicare data from CSV file
Output: Measure Accuracy Score
1. Select the features using F_Score
 # SelectKBest() is a function under
 # feature_selection under sklearn library
 # f_classif uses Anova F-value for classification
 # purposes
2. selec_features ← SelectKBest(f_classif, k = 10)
3..Remove the outliers using Z_Score
 # zscore a function available in Scipy python
 # package under stats module
4..z ← np.abs(stats.zscore(data))
5..Compute the Skewness to determine normal distribution of values
 #Pandas library in Python to measure unbiased
 #skewness.
6.skw ← data.skew()
7. Remove the outliers by identifying anything that is not the range of lower and upper bound
IQR ← IQR3 – IQR1
l_bound ← IQR1 - (IQR * 1.5)
u_bound ← IQR3 + (IQR * 1.5)
8. Assign X to columns and Y to target
9. Split X and Y into training and testing dataset in the ratio 80 to 20%
10. Train the models (Logistic,SVM and DLANN Model)
11. Predict the target for the above models.
12. Compute K-Fold accuracy for the models
# KFold from sklearn library will split data into 10
# folds where 9 folds are used for training and
# 1 fold for validation in an iterative manner;
# random state=7 is seed for random number
# generator
13. kfold ← KFold(n_splits=10, random_state=7)
14. Compute confusion matrix metrics and ROC for the above models.
15. Plot the area under receiver operating characteristic curve from the metrics module under sklearn library
16. auc_score ← metrics.roc_auc_score(y_test, y_pred_prob)
```

Algorithm 1.

Accuracy of the classification model [32] is determined in the present study by correctness of the confusion matrix and is as given in Equation 1.

$$Accuracy_{model} = \frac{(S_{TP} + S_{TN})}{S_{Total}} \quad (1)$$

where $Accuracy_{model}$ gives the classification accuracy. A higher accuracy of 99% is good but at times it also depends on the dataset.

Precision of the classification model gives the percentage the correct results among all the returned results and is as given in Equation 2

$$Precision_{model} = \frac{S_{TP}}{S_{TP} + S_{FP}} \quad (2)$$

where $Precision_{model}$ gives precision of a machine learning model for classification problem

	Predicted	
Actual	TP	FP
	FN	TN

Table 3: Layout of confusion matrix.

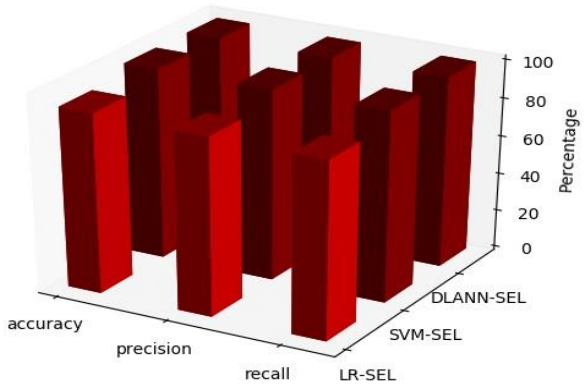

Figure 2: Confusion matrix metrics forl logistic regression, SVM and DLANN for selected feature dataset.

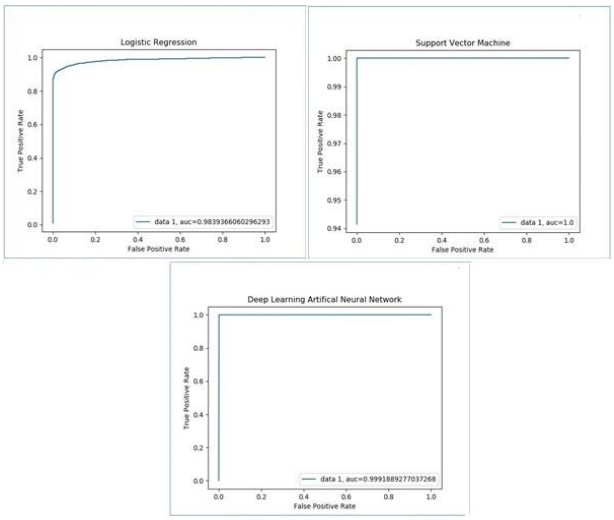

Figure 3: AUC for logistic regression, SVM and DLANN for selected feature dataset.

Recall is the capacity of the model to find data points of interest and is as given in Equation 3

$$Recall_{model} = \frac{S_{TP}}{S_{TP} + S_{FN}} \quad (3)$$

where $Recall_{model}$ gives the correct classification of positive samples by the machine learning model for the given binary classification problem.

One of the limitations of accuracy is its constraints in terms of test sample size which in our experiments has been considered as 20%. Thus, for a binary classifier as in our experiments, where we have pitted true positives against false negatives; Area Under Curve (AUC) gives a more generic approach as it evaluates the binary classifier model for random guesses. Thus, AUC provides a better perceived measure as compared to accuracy which is more tightly coupled to a threshold. In an event when accuracy cannot be used to clearly distinguish machine learning models AUC can work as an alternative deciding parameter [33]. The experimentation conducted provided K-Fold validation scores of 94.10% and 99.97% for logistic regression and SVM respectively.

Thus, the K-Fold accuracy of SVM is superior to Logistic Regression by 12.15%. We now consider the confusion matrix metrics for the selected feature dataset as illustrated in Figure 2.

It is further observed in Figure 2 that SVM was the top performer and marginally outscored DLANN which is an interesting observation which needs to be analysed further.

Figure 3 shows AUC for Logistic Regression, SVM and DLANN. From Figure 3 it is observed that SVM has the highest AUC of 1.0 followed by DLANN with an AUC of 0.99. The AUC of Logistic Regression is relatively least with a score of 0.98.

3.2 Computational time analysis

Based on the observations made from the binary classifier model it becomes imperative that apart from scoring high on accuracy and other associated metrics computational efficiency on resource constrained IoT environment is a necessary attribute for a low-cost data analysis system. Therefore, the investigators decided to compare the computational time of each model used for analysis. The hardware platform used for this aspect of analysis was a Raspberry Pi with Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM. The results of this analysis is as illustrated in Table 4.

As mentioned before, the application of feature selection and removal of outliers led to the reduction of dataset size from 8.7 MB to 2.9 MB. Therefore, it is common sense that for a dataset with selected variables the computational time will be naturally lower. This is of significance for resource constrained environments IoT environments such as the Raspberry Pi. It is observed under dataset with selected variables Logistic Regression outperforms SVM by 99.04% and DLANN by 98.02%. This clearly indicates that Logistic Regression is most computationally efficient as compared to SVM and DLANN. Also, SVM outperformed DLANN and Logistic Regression in terms of AUC, confusion matrix metrics and

Binary classifier Model	Computational Time in seconds
	Raspberry Pi B
Logistic Regression – Selected Dataset	37.81
SVM – Selected Dataset	3949.02
DLANN – Selected Dataset	1918.59

Table 4: Computational time of machine learning and deep learning models.

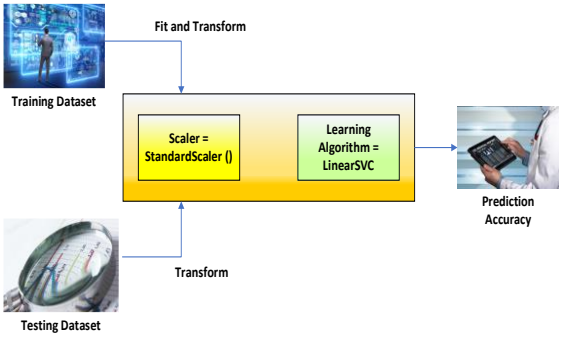

Figure 4: Pipeline architecture for SVM on Raspberry Pi.

```
Output: Pipelined Architecture of SVM
1.pipe_lrSVC ← Pipeline([('scaler', StandardScaler()), ('clf', LinearSVC())]) #Build the pipeline
2. r ← pipe_lrSVC.fit(X_train, y_train)
3. y_pred ← pipe_lrSVC.predict(X_test) #predict
```

Algorithm 2

```
Input : newmeddata.csv # The original dataset
Output: noisy_data.csv # The noisy dataset
1. σ ← 0.1 # standard deviation is 0.1
2. μ ←0 # mean is 0
3. noise ← actual_data + σ * random (size (actual_data)) + μ
4. noisy_data.csv ← actual_data + noise
#noisy_data.csv is the data with added Gaussian noise
5. target_variable ← int (actual_target_variable + noise)
```

Algorithm 3

K-Fold validation tests. This motivated the investigators for further analysis where they built a model where SVM provides robust performance and also is computationally time efficient.

3.3 Pipelined support vector machine architecture and Gaussian noise

Pipeline allows us to fit a model by combining a number of transformations and executing a predictor once. The software pipeline architecture as provided by scikit-learn [24] is as illustrated in Figure. 4.

In Python the Pipeline class [34] allows the collation of multiple processes into a single estimator. Therefore, we can fit the pipeline to the whole training data and also transform it to test data without the need for doing the same individually. Linear Support Vector Classification abbreviated as LinearSVC uses a linear kernel, is faster and can also scale rapidly. These parameters were fed to the pipeline to reduce the computational time required for SVM on raspberry pi.

The algorithm implemented in our model of pipelined SVM is as illustrated in Algorithm 2.

Here Gaussian noise is added to the dataset to benchmark the performance of SVM against Logistic Regression and Deep Learning Artificial Neural Network. The presence of Additive Gaussian Noise [35][36] is known to have impact on the distribution of the data. To check the robustness of the different classifier models a common data corruption technique through Gaussian noise was applied. Many such analysis were conducted in [37] to benchmark neural network robustness. In our work the noise signal was set with mean 0 and standard deviation of 0.1.To simulate the Gaussian Noise the NumPy Random Normal function was used, which generates values from the Gaussian distribution. The values assumed for μ was 0 and σ = 0.1. The additive noise is as generalized [38] in Equation 4.

$$M_{Rno,Fno} = O_{Rno,Fno} + \epsilon_{Rno,Fno} \quad (4)$$

where $M_{Rno,Fno}$ is the modified data point; $O_{Rno,Fno}$ is the original data point and $\epsilon_{Rno,Fno}$ is the random noise approximately equal to the distribution (μ, σ^2); where μ is mean and σ^2 is the variance. The algorithm for Gaussian Noise implementation is illustrated in Algorithm 3.

The analysis of K-Fold validation tests came with a result of 78.88% for Logistic Regression and 78.58% for SVM which indicated the similar performance of both the models in presence of Gaussian Noise. The performance of Logistic Regression dropped by 15.22% and performance of SVM dropped by 21.39 %. This clearly indicates that in the presence of noise logistic regression performed at an acceptable level. We further continued our experiments for results with confusion metrics as illustrated in Figure 5.

From Figure 5 it is observed that the performance of SVM in terms of accuracy is least 58.44% in presence of Gaussian Noise. The precision of Logistic Regression and DLANN was good and exhibited similar performances of 49.79%. and 50.79%. However, it could be observed that the precision was one of the worst affected metrics and the performance for Logistic Regression dropped by 46.8%, SVM by 66.53%, and DLANN by 49.11%. This performance was compared with performances of machine learning models run on dataset with selected features. A low precision for SVM could basically indicate a large number of false positives. On the contrary, a high value of recall of 99.16% indicates that SVM was very sensitive and could successfully identify true positive observations. The analysis was continued for AUC metric.

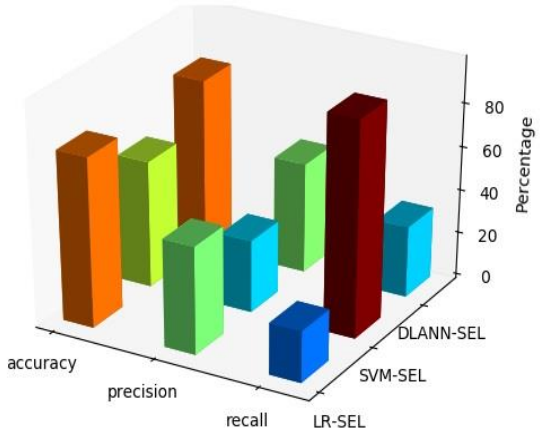

Figure 5: Confusion matrix metrics for logistic regression, SVM and DLANN for selected feature dataset and with gaussian noise.

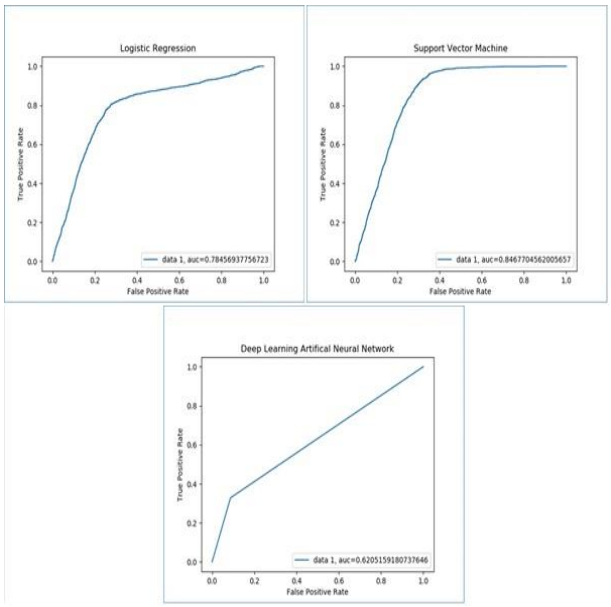

Figure 6: AUC for logistic regression, SVM and DLANN for selected feature dataset with gaussian noise.

Binary classifier Model	**Computational Time in seconds**
	Raspberry Pi B
Logistic Regression – Selected Dataset with Gaussian Noise	23.57
SVM – Selected Dataset with Gaussian Noise	382.34
DLANN – Selected Dataset – Gaussian Noise	1887.36

Table 5: Computational time of machine learning and deep learning models.

From Figure 6 the investigators observe that the performance of SVM is better compared to other models. It outscores Logistic Regression by 7.2%, and DLANN model by 22.65%.

3.3.1 Computational time analysis

As indicated in the introduction section the investigators performed the computational time analysis of different methods. This computational time analysis is illustrated in Table 5.

Here we observe that Logistic Regression was the most computationally efficient in terms of execution time. However, with a pipeline SVM outperformed its nearest competitor DLANN by 79.7 4% and was inferior to Logistic Regression by 93.83%. Therefore, SVM improved its performance in terms of computational execution time. Additionally, it was observed that in presence of Gaussian Noise, the accuracy of most of the models dropped and DLANN emerged as slight winner with little bit of consistency and SVM exhibited low recall and high precision thereby exhibiting its fitness for the dataset under consideration. Also, the proposed Pipelined model of SVM achieved a better performance in terms of computational time to its nearest competitor the DLANN model.

4 Discussion

The investigators in the present work implemented a pipeline SVM model to test it against known benchmarks of Logistic Regression and Deep Learning Neural Network for performance optimization in terms of computational time and accuracy metrics for a resource constrained environment of Raspberry Pi. Therefore, the investigators explored statistical technique of F Score for feature selection and could shortlist top 10 features. The investigators further processed outliers by applying Inter Quartile Range. This helped the investigators to balance the skewness of the data. Thus, the modified dataset with reduced storage requirements was tested on Raspberry PI for machine learning models like logistic regression, SVM and DLANN for binary classification and performance benchmarking. K-Fold accuracy of SVM was superior to Logistic Regression by 12.15%. Confusion matrix metrics where further applied to test the machine learning models and SVM achieved better performance and at times was at par with Deep Learning Neural Network. The uniqueness of the present work is that it dealt with the training time that SVM takes which is usually large. Thus reducing training time was of paramount importance as the platform were, SVM was to be implemented was Raspberry Pi. This was achieved by implementing SVM with a pipelined architecture. Thus SVM achieved a better performance in terms of computational time to its nearest competitor the DLANN model by 79.74%. .SVM is prone to noise, thus the optimized and pipelined architecture of SVM was benchmarked with Deep Learning in the presence of Gaussian noise. The accuracy of most of the models dropped and DLANN emerged as slight winner with little bit of consistency and SVM exhibited low recall and high precision thereby exhibiting its fitness for the dataset under consideration. The better accuracy of DLANN with selected features and under noise may be attributed to the fact that noise could have added as a regularization factor thus boosting the performance of DLANN. This clearly provides some pathway for future work in terms of

extending pipeline architectures for Deep Learning algorithms [39],[40],[41], which are efficient but slow and are visualized for working in resource constrained environments of IoT.

4.1 Limitations of the present work

The analysis was considered for a single medical dataset. In future the capabilities of the models could be generalized for a range of datasets. With parallel environments for machine learning models and with IoT clusters based on graphical processing units (GPU's) for remote computing the models could be made much more computationally feasible. Also, techniques like PCA for feature selection and its interaction for deep learning algorithms was not explored in the present work.

5 Conclusion

Overall, the investigators conclude that SVM exhibited its robustness in terms of relatively good performances for all computational setups of optimized, and corrupted datasets in resource constrained environments of IoT. The impact of additive noise had distressing effects on most models and may be a concern in an environment where devices collect data from sensors. As stated, the analysis was conducted on a single dataset thereby limiting the validation of the conclusions derived. The feature selection of dataset resulted in reduction of dataset size by 67% but had a minor loss in terms of accuracy of the classifier models like Logistic Regression, SVM and DLANN. Therefore, we can safely suggest that SVM had a relatively stable performance across all the scenarios and at times was better than DLANN model. Additionally, we suggest that pipeline architectures and automating machine learning models had a good impact on resource constrained environments like Raspberry Pi. SVM pipelined model outscored DLANN model by 79.94% for a featured selected and Gaussian noise added dataset in terms of computational time. Thereby, the investigators have concluded SVM as the model of choice for analysing similar datasets. Therefore, the core contributions of this work were: i) implementing a pipelined Support Vector Machine model for performance benchmarking against Logistic Regression and Deep Learning Neural Network for computational time efficiency and accuracy metric for a relatively largest dataset, and ii) a brief analysis of computational time analysis for these general methods for SVM using Raspberry Pi. In future, the investigators would like to explore how the machine learning and deep learning models that can detect noise and outliers and automatically improve their learning abilities for complex pipelined models, in a constrained environment of an IoT device enabled by Graphics Processing Unit (GPU).

Acknowledgments

The authors would like to thank the School of Global Health Management and Informatics for the permission to use the University of Central Florida (UCF), Decision Support Systems and Informatics Laboratory facilities to conduct the research work and related documentation.

Research Study	Analysis Techniques	Results
This project	Pipelined Support Vector Machine, Logistic Regression and Deep Learning Artificial Neural Network on Raspberry Pi environment	Pipelined SVM achieved a better performance in terms of computational time measurement to its nearest competitor the DLANN model by 79.74%.
Sheets et.al, 2017 [17]	Combination of contrast mining and Logistic Regression was used.	Electronic Health Record (EHR) contrast mining with Logistic Regression predicted 5% of patients contributing to 50% of healthcare expenses.
Nalepa & Kawulok J., 2018 [9]	Trained Support Vector Machine for large datasets with different kernels.	SVM has been successful in solving a variety of pattern recognition tasks; its main drawbacks were the huge time and memory related complexities.
Sakr et.al, 2016 [18]	Deep Learning Convolutional Neural Network (CNN) and Support Vector Machine	SVM model achieved high classification accuracy of 94.8% while CNN could achieve 83%
Pei et.al., 2021 [21]	Deep Learning Convolutional Neural Network (CNN) and White Gaussian Noise	Classification performance of Deep Learning CNN drops significantly when noise is added.

Table 6: Comparison of research projects and analysis methods.

References

[1] Allhoffa F. & Henschke A (2018). The Internet of Things: Foundational ethical issues, *Internet of Things*, pp. 55–66. https://doi.org/10.1016/j.iot.2018.08.005

[2] Haller S., Karnouskos S., & Schroth C (2009). "The Internet of Things in an Enterprise Context," in *Future Internet – FIS 2008, Lecture Notes in Computer Science*, vol. 5468, pp 14-28. https://doi.org/10.1007/978-3-642-00985-3_2

[3] Zhang Z-K., Cho M , Wang C-W.., Hsu C-W,Chen C-K, & Shieh S (2014). IoT Security: Ongoing Challenges and Research Opportunities,

Proceedings of the 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 230-234. https://doi.org/10.1109/SOCA.2014.58

[4] Gokhale P., Bhat O., Bhat S (2018). Introduction to IOT, *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5(1), pp. 41- 44. https://doi.org/10.17148/iarjset.2018.517

[5] Wan T.T.H, Gurupur V (2020). Understanding the Difference between Healthcare Informatics and Healthcare Data Analytics in the Present State of Health Care Management, *Health Services Research & Managerial Epidemiology*, vol. 7, pp. 1-3. http://dx.doi.org/10.1177/2333392820952668

[6] Shukla S., Hassan M.F., Khan M.K., Jung L.T., Awang A (2019).An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment, *PLoS ONE*, pp.1-31. http://dx.doi.org/10.1371/journal.pone.0224934

[7] Noble W.S (2006). What is a support vector machine? *Nature Biotechnology*, Vol.24, pp. 1565–1567. https://doi.org/10.1038/nbt1206-1565

[8] Bradley A.P (1997).The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms, *Pattern Recognition*, vol. 30(7), pp. 1145-1159. https://doi.org/10.1016/S0031-3203(96)00142-2

[9] Nalepa J. , Kawulok M (2019). Selecting training sets for support vector machines: a review. *Artif Intell Rev* 52, pp. 857–900. https://doi.org/10.1007/s10462-017-9611-1

[10] Papadonikolakis M., Bouganis C. & Constantinides G (2009). "Performance comparison of GPU and FPGA architectures for the SVM training problem," *2009 International Conference on Field-Programmable Technology*, pp. 388-391. https://doi.org/10.1109/FPT.2009.5377653

[11] Huang C-L, Chen M-C, Wang C-J (2007). Credit scoring with a data mining approach based on support vector machines, *Expert Systems with Applications*, vol. 33, pp. 847–856 https://doi.org/10.1016/j.eswa.2006.07.007

[12] Yazici M T. , Basurra S. & .Gaber M M (2018). Edge Machine Learning: Enabling Smart Internet of Things Applications, *Big Data and Cognitive Computing*, vol. 2: 26, pp. 1-17. https://doi.org/10.3390/bdcc2030026

[13] Nguyen M H. Torre F de la (2010). Optimal feature selection for support vector machines, *Pattern Recognition*, vol.43, pp. 584–591 https://doi.org/10.1016/j.patcog.2009.09.003

[14] Sanz H, Valim C., Vegas E, Oller J M. & Reverter F (2018). SVM-RFE: selection and visualization of the most relevant features through non-linear kernels, *BMC Bioinformatics*, vol. 19:432, pp 1-18. https://doi.org/10.1186/s12859-018-2451-4

[15] Gurupur V. P, Kulkarni S. A., Liu X., Desai U., & Nasir A (2018). Analysing the power of deep learning techniques over the traditional methods using medicare utilisation and provider data, *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 99-115. https://doi.org/10.1080/0952813X.2018.1518999

[16] Zardo P., Collie A (2014). Predicting research use in a public health policy environment: results of a logistic regression analysis, *Implementation Science*, vol. 9, pp. 1-10. https://doi.org/10.1186/s13012-014-0142-8

[17] Sheets L., Petroski G.F., Zhuang Y., Phinney M.A,. Ge B, Parker J.C., Shyu C-R (2017). Combining Contrast Mining with Logistic Regression to Predict Healthcare Utilization in a Managed Care Population, *Applied Clinical Informatics*, vol. 8: 2, pp. 430-446. https://doi.org/10.4338/aci-2016-05-ra-0078

[18] Sakr G. E, Mokbel M., Darwich A., Khneisser M. N & Hadi A (2016). Comparing deep learning and support vector machines for autonomous waste sorting, *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pp. 207-212. https://doi.org/10.1109/IMCET.2016.7777453

[19] Ravi D., Wong C., Deligianni F., Berthelot M., Andreu-Perez J., Lo B., &. Yang G-Z (2017). Deep Learning for Health Informatics, *IEEE Journal of Biomedical and Health Informatics*, vol. 21: (1), pp.4-21. https://doi.org/10.1109/jbhi.2016.2636665

[20] Gangsar P. & Tiwari R (2018). Effect of noise on support vector machine based fault diagnosis of IM using vibration and current signatures, *MATEC Web of Conferences*, vol. 211. http://dx.doi.org/10.1051/matecconf/201821103009

[21] Pei Y., Huang Y., Zou Q., Zhang X. & Wang S (2021). Effects of Image Degradation and Degradation Removal to CNN-Based Image Classification," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43: 4, pp. 1239-1253. https://doi.org/10.1109/TPAMI.2019.2950923

[22] Wu X. & Zhu X (2008). Mining with Noise Knowledge: Error-Aware Data Mining, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.38: (4), pp.15-19. https://doi.org/10.1109/CIS.2007.7

[23] Zualkernan A., Zualkernan I A., Dhou S, Judas J, Sajun A R, Gomez B R., Hussain L A., Sakhnini D (2020), Towards an IoT-based Deep Learning Architecture for Camera Trap Image Classification, *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*,pp. 1-6. https://doi.org/10.1109/GCAIoT51063.2020.9345858

[24] Scikit-learn Machine Learning in Python. [Online]. Available: https://scikit-learn.org/ stable/

[25] Tukey J (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading MA

[26] Zhao Q., Zhou G., Zhang L., Cichocki A. & Amari S (2016).Bayesian Robust Tensor Factorization for

Incomplete Multiway Data, *IEEE Transactions on Neural Networks and Learning Systems*, vol.27:(4), pp.736-748
http://dx.doi.org/10.1109/TNNLS.2015.2423694

[27] Khan Z., Naeem M., Khalil U., Khan D. M., Aldahmani S. & Hamraz M (2019). Feature Selection for Binary Classification Within Functional Genomics Experiments via Interquartile Range and Clustering, *IEEE Access*, vol. 7, pp.78159-78169.
https://doi.org/10.1109/ACCESS.2019.2922432

[28] Yusoff S. B. & Wah Y. B (2012). Comparison of conventional measures of skewness and kurtosis for small sample size, *2012 International Conference on Statistics in Science, Business and Engineering* (ICSSBE), pp.1-6.
https://doi.org/10.1109/ICSSBE.2012.6396619

[29] Heymann S., Latapy M. & Magnien C (2012). Outskewer: Using Skewness to Spot Outliers in Samples and Time Series, *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp.527-534.
https://doi.org/10.1109/ASONAM.2012.91

[30] Xu L., Hu O., Guo Y., Zhang M., Lu D., Cai C. B., Xie S., Goodarzi M., Fu H. Y., She Y. B (2018). Representative splitting cross validation, *Chemometrics and Intelligent Laboratory Systems*, vol.183, pp.29-35.
https://doi.org/10.1016/j.chemolab.2018.10.008

[31] Tharwat A (2018). Classification assessment methods, *Applied Computing and Informatics*, pp.1-13.
https://doi.org/10.1016/j.aci.2018.08.003

[32] Fatourechi M., Ward R. K., Mason S. G., Huggins J., Schlög A., & Birch G. E (2008). Comparison of Evaluation Metrics in Classification Applications with Imbalanced Datasets, *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, pp.777 – 782.
https://doi.org/10.1109/ICMLA.2008.34

[33] Huang J. & Ling C (2005). Using AUC and Accuracy in Evaluating Learning Algorithms, *IEEE Transactions on Knowledge & Data Engineering*, vol.17:(3), pp.299-310.
https://doi.org/10.1109/TKDE.2005.50

[34] Pipelines and composite estimators, https://scikit-learn.org/stable/modules/compose.html

[35] Nadarajah S. & Kotz S (2007). On the Generation of Gaussian Noise, *IEEE Transactions on Signal Processing*, vol. 55 (3), pp.1172-1172.
http://dx.doi.org/10.1109/TSP.2006.888061

[36] Zhuang L. & Ng M. K (2020). Hyperspectral Mixed Noise Removal By ℓ1-Norm-Based Subspace Representation, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.13 , pp.1143-1157.
https://doi.org/10.1109/JSTARS.2020.2979801

[37] Hendrycks D., & Dietterich T. G (2018). Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations, *arXiv: Learning*, pp.1-13
https://arxiv.org/abs/1807.01697v5

[38] Domingo-Ferrer J., Seb´e F., & Castell`a-Roca J (2004). On the Security of Noise Addition for Privacy in Statistical Databases, *International Workshop on Privacy in Statistical Databases*, pp.149-161.
http://dx.doi.org/10.1007/978-3-540-25955-8_12

[39] Yao S., Zhao Y., Zhang A., Hu S., Shao H., Zhang C., Su L., Abdelzaher T (2018). Deep Learning for the Internet of Things, *Computer*, vol. 51: 5, pp. 32-41.
https://doi.org/10.1109/MC.2018.2381131

[40] Ma X., Yao T., Hu M., Dong Y., Liu W., Wang F., Liu J (2019).A Survey on Deep Learning Empowered IoT Applications, in *IEEE Access*, vol. 7, pp. 181721-181732.
https://doi.org/10.1109/ACCESS.2019.2958962

[41] Ahmed I., Din S., Jeon G., Piccialli F (2020). Exploring Deep Learning Models for Overhead View Multiple Object Text of the second section, in *IEEE Internet of Things Journal*, vol. 7: 7, pp. 5737-5744.
http://dx.doi.org/10.1109/JIOT.2019.2951365