Design and Implementation of a System for Comparative Analysis of Learning Architectures for Churn Prediction

Muhammad Usman, Waseem Ahmad, and Alvis Fong

The authors present a system for comparative analysis of learning architectures. Two benchmarked datasets, Cell2Cell and KDD Cup, serve as a use case of our system to provide insights on the extent of improvement DL can bring over classical ML. Four popular evaluation measures are used to compare the performance of popular DL architectures. The authors' experiments found that convolutional neural networks gave the best results in both use cases.

Abstract

Telecom companies are increasing their efforts in customer retention because acquiring new customers often costs much more than retaining existing ones. Therefore, it is important for operators to predict customer churns rapidly and accurately. Machine learning (ML) has been widely used for predictive churn modeling. However, classical ML methods require manual feature selection and time-consuming data preprocessing steps. To overcome these limitations, there is a paradigm shift toward deep learning (DL) for predicting churners. Although DL appears to be promising, the existing literature lacks comparative analysis of ML and DL techniques using benchmark churn datasets. Additionally, various DL architectures must be empirically evaluated to determine which type works best on churn data. We present a system for comparative analysis of learning architectures. Two benchmarked datasets, Cell2Cell and KDD Cup, serve as a use case of our system to provide insights on the extent of improvement DL can bring over classical ML. Four popular evaluation measures are used to compare the performance of popular DL architectures. Our experiments found that convolutional neural networks gave the best results in both use cases.

Introduction

In the competitive telecommunication market [1, 2], telecom operators are changing their strategy from a business model based on product strategy to one based on customer strategy [3]. Churn prediction, which aims to identify customers likely to switch to another company before they do, can give operators a competitive edge in customer retention.

A real-world churn dataset normally consists of hundreds of millions of transactions. It is becoming a challenge to perform predictive analytics on such massive datasets using classical machine learning (ML) algorithms [4]. Recently, deep learning (DL) methods have been used to predict customer churn using data from various sources [5]. These methods are suitable for large volumes of data.

Data representation plays an important role in DL methods. For example, Karanovic *et al.* [5] use one-dimensional image churn data (KDD Cup) to build classification model with convolutional neural networks (CNNs). Data can also be represented as a sequence of timestamped events [6]. CNNs and recurrent neural networks (RNNs) are among the popular DL models for classifying churn data. These models can be further divided based on their input data representation, architecture, and other network parameters. However, no study has been carried out to compare the prediction performance of various DL methods on churn datasets. Churn datasets are often imbalanced with churners being the minority group. Imbalanced datasets can make effective learning difficult.

We present a system for comparative analysis of DL architectures with churn data in different representations. Image data are processed using CNNs, and sequence data are processed using RNNs. Our aim is to compare the classification performance of CNNs and RNNs using different data representations. The purpose is to present empirical results on which DL method works best on benchmarked churn datasets in order to lay the foundation for future generalizability analysis to other churn datasets. In addition, we also compare the performance of DL models with popular classical ML methods to demonstrate the superiority of DL over ML.

Related Work

ML has been applied to the churn prediction problem for over a decade. For example, Huang *et al.* [7], introduced a new set of features using new window techniques by using one and two predicators. The features with their proposed new window techniques were found to be efficient for churn prediction in landline telecommunication. However, their proposed techniques were not properly justified, and the reasons for choosing specific classifiers and evaluation methods were not given.

In [8], the authors aimed to discover the relationship between categorical values of features and class to facilitate feature selection. Their approach was found to be effective using four classifiers: decision trees (DTs), naïve Bayes (NB), logistic regression (LR), and support vector machine (SVM). Limitations of the work include missing details of features and feature sets used for training and testing.

Digital Object Identifier:
10.1109/MCOM.110.2100145

Muhammad Usman is with the Shaheed Zulfikar Ali Bhutto Institute of Science & Technology; Waseem Ahmad is with Toi-Ohomai Institute of Technology;Alvis Fong is with Western Michigan University.

0163-6804/21/$25.00 © 2021 IEEE

In [5], the authors proposed classification by rule learning (CRL), which consists of two steps: first, generating rules, and second, predicting the desired category according to the rules. It was reported that results were not always as expected, and this model might not be suitable for larger datasets. A critical limitation of the work is that the number of instances used for training and testing models were not sufficiently large for realistic churn data.

Ullah *et al.* [9] combined classification with clustering to identify the churn customers and provided the factors behind their observations. Feature selection was performed by using information gain and correlation attribute ranking filter. Their model first classifies churn customers data using Random Forest (RF). Their algorithm then clusters the churning data by using cosine similarity to provide group-based retention offers.

Qureshi *et al.* [3] applied well-known algorithms of regression analysis, DT, and neural networks (NNs) for churn prediction. Unfortunately, the authors did not provide reasons for selecting p-value and correlations as feature selection techniques. In resampling, the majority class comprising non-churners was under-ied up 150 percent of minority class (churners). In their study, the number of churners was 6231. Hence, the whole dataset had fewer than 15,600 instances for training (70 percent) and testing (30 percent). These numbers of instances are not adequate for realistic churn analysis. Moreover, their reasons for choosing specific classifiers and evaluation methods were missing.

Ahmad *et al.* [10] considered four algorithms: DT, RF, Gradient Boosted Machine Tree (GBM), and Extreme Gradient Boosting (XGBoost). They found that XGBoost gave the best result when measured using the area under the curve (AUC) metric. The authors used customer social network in the prediction model by extracting social network analysis (SNA) features. The use of SNA improved the AUC performance of their model from 84 percent to 93.3 percent.

Key limitations of classical ML methods include the need for feature selection and difficulty in handling big churn datasets. More recently, researchers have applied DL to the churn prediction problem [5, 11, 12]. Prashanth *et al.* [13] wrote a short paper that compared linear (LR) against nonlinear (RF and DL) models for churn prediction. They found that the latter performed best. However, critical comparative analysis of DL for churn prediction using benchmark datasets is lacking. This represents an important knowledge gap. It is very challenging for churn prediction systems to mine massive data. In this article, we propose a system that provides flexibility in facilitating comparative analysis of different ML/DL classifiers. Use cases based on benchmark datasets allow us to validate the system. Furthermore, its flexibility extends beyond just comparison of different classifiers (e.g. handling of data imbalance).

System Architecture

Churn prediction is a multi-stage process. Figure 1 shows our 5-stage system architecture. Datasets are gathered at Stage-1. Stage-2 selects a research focus (e.g., comparative analysis of classifiers and/or treatment of data imbalance). Stage-3 is data preprocessing, while classification techniques are selected and applied at Stage-4. Finally, Stage-5 shows the results through evaluation.

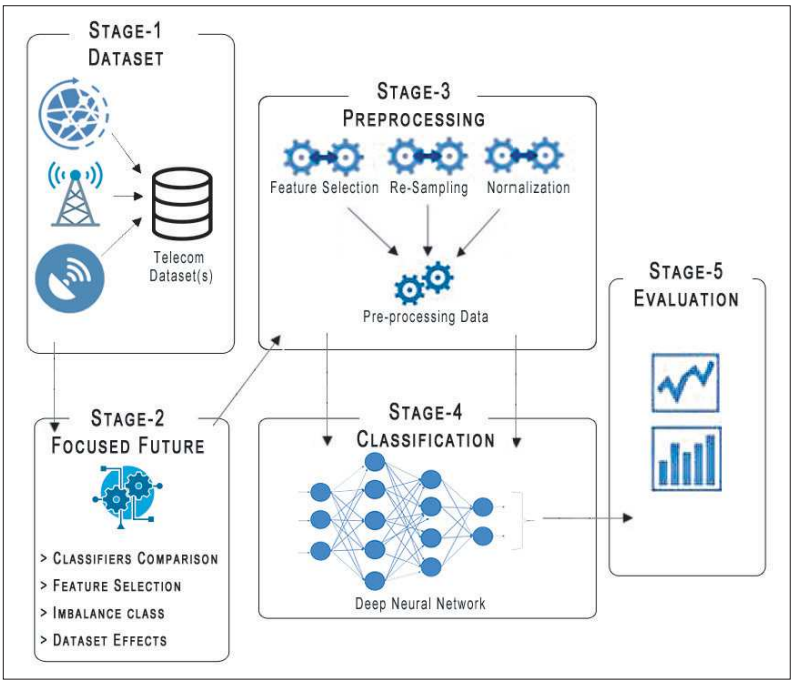

FIGURE 1. System architecture for implementation.

Stage-1: We begin by selecting suitable benchmark datasets as use cases. To enhance generalizability of findings, it is necessary to include datasets with different numbers of instances and attributes, and a diverse mix of categorical and numeric features. Cell2cell and KDD Cup Churn datasets are widely used in customer churn modeling and prediction. The first dataset has a profound imbalance of class labels (7.34 percent churners vs. 92.66 percent non-churners). The second dataset is relatively less imbalanced with churn instances making up 29 percent.

Stage-2: The effectiveness and performance of various classifiers are measured and critically evaluated (i.e., comparing various classifiers).

Stage-3: Once the focus area is defined, we need to perform the necessary transformation and filtering of each dataset to facilitate subsequent classification. Popular data preprocessing measures include data normalization, outlier removal, and missing data management. Specifically, the Cell2Cell dataset originally had 77 features, where two categorical features were removed due to the presence of many missing values. Apart from those features, all the missing values of the remaining 75 features were replaced with the maximum value of that column. The categorical features were transformed to numeric features using one hot encoding, where each category of the feature was mapped to a vector containing 1 or 0 depending on the presence or absence of that category. This made the number of vectors equal the number of categories in the features. The dataset was normalized to a value between 0 and 1 to avoid scale biasness in the final model.

The KDD Cup dataset originally had 231 features with 18 features having no data entries. Those features were removed from the data-

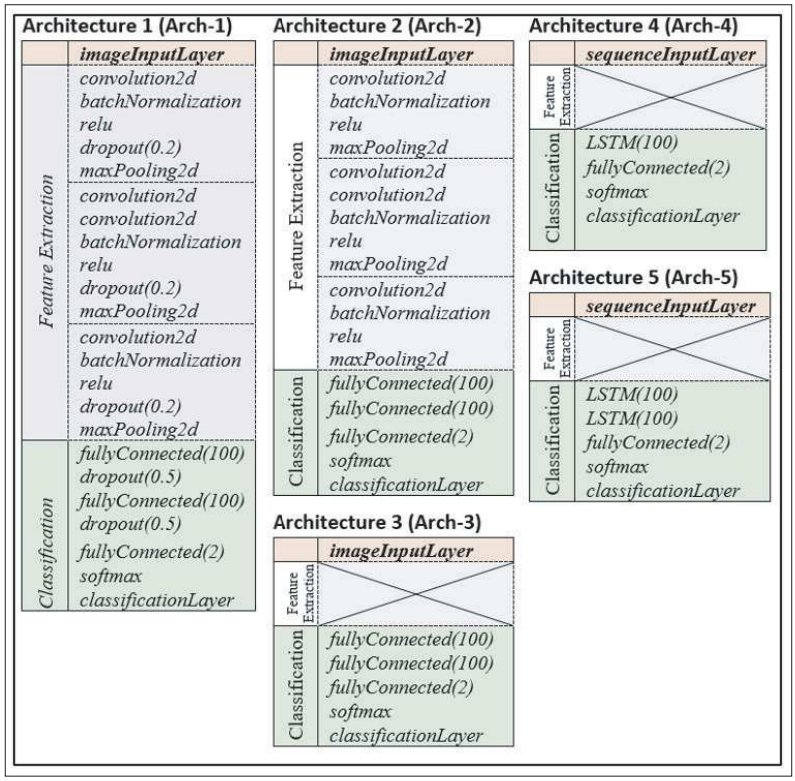

FIGURE 2. Deep learning architectures evaluated.

set. Moreover, the dataset had several features with missing values. A missing value threshold of 75 percent, which was found empirically to be optimal, was applied to further filter the feature space. After data feature removal, a final dataset with 75 features was obtained. The categorical features in the dataset were transformed into numeric features using frequency encoding. The transformation strategy was selected because several categorical features had >100 categories, meaning one hot encoding could not be implemented. Using frequency encoding, category values in a feature were replaced with its respective frequencies. Finally, the dataset was normalized to a value between 0 and 1 to keep uninform feature weights in model construction.

Stage-4: Here, we compare classical ML methods to contemporary DL. The classical ML methods chosen for this study are decision tree learning (J48), Bayesian networks (BNs), naïve Bayes (NB), and multi-layer perceptrons (shallow NNs). These methods are widely used and are well understood in the research community [14].

For DL, we experiment with two ways of data representation:

1. Image data
2. Sequence data

CNNs are used to learn patterns from the image data as in [5], and RNNs in the form of long short-term memory (LSTM) are used to learn from sequence data [15]. Specifically, we analyze four cases:

Case 1: CNN on one-dimensional image data
Case 2: CNN on multi-dimensional image data
Case 3: LSTM with one hidden layer
Case 4: LSTM with two hidden layers

In case 1, each row (data instance) is considered as a single (row) image, which is used to construct a predictive DL model. In case 2, each row (data instance) is decomposed into an $n \times m$ matrix. The resulting $n \times m$ matrix is considered as an image and is used to train the CNN. In cases 3 and 4, LSTM is used, incorporating one or two hidden layers of neurons, respectively.

Stage-5: Classifier evaluation. For benchmarking, we adopt four widely used evaluation measures: accuracy, area under the curve (AUC), G-Mean, and average [7–10]. Accuracy is derived from the sum of true positive (TP) and false negative (FN) divided by the total number of instances. G-Mean is derived from sensitivity and specificity, which are in turn derived from TP, FN, and false positive (FP) and true negative (TN). AUC is derived from the receiver operating characteristic (ROC) curve. AUC represents a model's capability to distinguish between various classes in the data. Average aggregates average, AUC, and G-Mean.

Use Case Analysis

Once the system in Fig. 1 was implemented, it was put into use to:

1. Investigate whether DL methods outperform classical ML methods on reasonably large benchmark churn datasets
2. Find which DL architecture gives better classification results

Figure 2 summarizes the five DL architectures that were used to evaluate the effectiveness of various DL methods on churn data. The main difference between Architectures 1 and 2 is the dropout layer. The dropout layer is used to overcome model overfitting by randomly setting outgoing edges of neurons to zero. The optimal dropout rates were found empirically. In both architectures (1 and 2), there are two main steps: feature extraction and classification. The layers of feature extraction consist of convolution, normalization, rectified linear unit function (relu) activation, max pooling, and dropout layers. In the classification step, two hidden layers, each consisting of 100 neurons, and an output layer consisting of two outputs (churn and non-churn) are used.

In Architecture 3, the input layer is directly connected to the hidden layers, and there are no feature extraction layers. The contribution of a feature extraction layer would be evaluated by comparing the outcomes of Architectures 1 and 2 with classification results of Architecture 3. Architectures 4 and 5 consist of LSTM layer/s, and feature extraction layers are also omitted. Architecture 4 has one hidden layer with 100 neurons, whereas Architecture 5 consists of two hidden layers, each having 100 neurons. Other parameter values used were: Max_Epoch = 300, Batch_Size = 10,000, Learning_Rate = 0.001, Optimizer = Adam, and Shuffle_After_Every_Epoch = true.

All the experiments involving classical ML and DL methods were carried out using Matlab 2020a. The results shown in Tables and Table 2 were averaged over five runs. In each run, the datasets were randomly divided into 70 percent training and 30 percent test data.

As shown in Table 1, DL Architecture 1 (Arch1), which has feature extraction and classification layers, produced the best overall result as measured using the Average metric, achieving a performance of nearly 77 percent. LSTM net-

works achieved poor classification results on the Cell2Cell dataset. Table 1 shows DL methods that use image data representation produced much better results than the sequence-based data representation (i.e., Cases 1 and 2 vs. Cases 3 and 4). Among the classical ML methods, the J48 algorithm produced the worst results, whereas Bayesian Net (BN) performed marginally better than NB and MLP.

The KDD Cup dataset has a much more pronounced class label imbalance than the Cell2Cell dataset. As Table 2 shows, we obtained quite interesting and diverse results on this dataset. From Table 2, it is evident that Architectures 3, 4, and 5 resulted in high accuracy and low G-Mean values, indicating that these architectures produced highly accurate results on positive labels but poor performance on negative labels (i.e., low TN rates). These results highlight the fact that the absence of the feature extraction layer in this experiment led to poor classification (high TP but low TN). Overall, 1D image data representation using Arch1 achieved the best results on the KDD cup dataset. On the other hand, NB achieved poor results on this dataset due to very poor classification accuracy.

In the case of imbalanced data, there is a trade-off between TP and TN rates. This means in order to obtain a high TN rate, we must lower the TP rates. This is evident when comparing the results of MLP (or J48) and BN. Among the DL methods, the results of Architectures 1 and 2 also echo the same trade-off phenomenon. Arch2 achieved a better G-Mean value than Arch1. However, this marginally better G-Mean value came at the cost of a relatively lower classification accuracy value. Also, Arch2 did not have any dropout layer, which could help reduce model overfitting. Better TN rates obtained using Arch2 suggest that mitigating model overfitting is critically important in imbalanced data to achieve robust modeling.

Discussion

Significance: As highlighted in the Related Work section above, there is a knowledge gap in critical comparative analysis of DL for churn prediction using benchmark datasets. This article aims to contribute toward filling this gap. Specifically, multiple DL configurations were tested on two benchmark datasets for a direct comparison. Analysis of the results shed light on the underlying reasons for the observed differences in performance.

Sustainability: After the system is deployed, it will be updated periodically through a maintenance program. Specifically, the proposed system will be further validated and finetuned with new datasets.

Limitations of study: The results presented represent analysis of use cases involving two benchmark datasets. The study should be broadened to validate the system configurations on more datasets.

Conclusion and Future Work

Our literature review reveals that customer churn prediction is an important task for the telecom industry. For the last few years, especially after worldwide deregulation of the telecommunications sector, customer churn activity is increasing throughout the world.

	Accuracy	AUC	G-Mean	Average
J48	0.738	0.666	0.664	0.702
BN	0.751	0.821	0.669	0.748
NB	0.722	0.792	0.739	0.743
MLP	0.743	0.810	0.688	0.746
Case1[a] Arch1	0.757	0.832	0.730	0.769
Case1[a] Arch3	0.754	0.831	0.682	0.755
Case2[b] Arch2	0.751	0.830	0.658	0.747
Case2[b] Arch3	0.753	0.827	0.709	0.760
Cases3,4[c] Arch1	0.728	0.806	0.502	0.691
Cases3,4[c] Arch4	0.727	0.803	0.552	0.702

[a] Case 1: 1-dimensional image data
[b] Case 2: $m \times n$-dimensional image data
[c] Cases 3 and 4: LSTM networks

TABLE 1. Results of classification on the cell2cell dataset.

	Accuracy	AUC	G-Mean	Average
J48	0.903	0.569	0.311	0.672
BN	0.724	0.653	0.597	0.675
NB	0.363	0.640	0.512	0.270
MLP	0.908	0.626	0.270	0.678
Case1[a] Arch1	0.914	0.639	0.271	0.685
Case1[a] Arch3	0.891	0.605	0.323	0.677
Case2[b] Arch2	0.911	0.617	0.230	0.667
Case2[b] Arch	0.927	0.708	0.069	0.658
Cases3,4[c] Arch1	0.891	0.644	0.094	0.630
Cases3,4[c] Arch4	0.893	0.660	0.092	0.635

[a] Case 1: 1-dimensional image data
[b] Case 2: m-by-n-dimensional image data
[c] Cases 3 and 4: LSTM networks

TABLE 2. Results of classification on the KDD Cup dataset.

Machine learning classifiers have been employed to predict the potential churn of customers with varying degrees of success. In this article, we have designed and implemented a multi-staged system for churn predictions. Our system consists of five stages:

1. Selection of datasets
2. Focused areas of research
3. Preprocessing
4. Choosing classification techniques
5. Evaluation

Moreover, this article has presented a thorough comparison of model performances of classical ML methods (Decision Tree Learning, Bayes Net, Naïve Bayes, and Multilayer Perceptrons), and DL architectures (convolutional neural networks and recurrent neutral networks). Evaluated on two benchmarked churn datasets (Cell2Cell and KDD Cup), our use case analysis of the implemented system shows that DL (especially CNNs) produced more effective classification models on churn datasets than others.

Possible future research directions include:

- In images, pixel positioning is important to obtain edges and boundaries of objects.

Our literature review revealed that customer churn prediction is an important task for the telecom industry. For the last few years, especially after worldwide deregulation of the telecommunications sector, customers churn activity is increasing throughout the world.

Therefore, it would be interesting to arrange the feature positioning of churn data based on some measure of feature importance or by applying feature weighting to intelligently rearrange the feature space. This rearrangement of features is expected to produce more robust and accurate classification models.

- More rigorous efforts are required to develop DL architectures that can maximize classification accuracy on churn data by experimenting with different arrangements of feature extraction layers and classification layers using various parameter settings of convolution, pooling, and dropout layers.

References

[1] G. S. Kuo, "Telecommunications Industry Markets: Vision and Potential," *IEEE Commun. Mag.*, vol. 36, no. 11, Nov. 1998, pp. 95–96.

[2] S. Bregni, M. Decina, and G. Bruzzi, "Traffic Trading in the Competitive International Voice Market," *IEEE Commun. Mag.*, vol. 47, no. 8, Aug. 2009, pp. 100–06.

[3] S. A. Qureshi *et al.*, "Telecommunication Subscribers' Churn Prediction Model Using Machine Learning," *8th Int'l. Conf. Digital Info. Management*, Islamabad, 2013, pp. 131–36. DOI: 10.1109/ICDIM.2013.6693977.

[4] N. Lu *et al.*, "A Customer Churn Prediction Model in Telecom Industry Using Boosting," *IEEE Trans. Industrial Informatics*, vol. 10, no. 2, May 2014, pp. 1659–65. DOI: 10.1109/TII.2012.2224355.

[5] M. Karanovic *et al.*, "Telecommunication Services Churn Prediction — Deep Learning Approach," *26th Telecommun. Forum*, Belgrade, 2018, pp. 420–25. DOI: 10.1109/TELFOR.2018.8612067.

[6] C.G. Mena *et al.*, "Churn Prediction with Sequential Data and Deep Neural Networks. A Comparative Analysis," 2019, arXiv preprint arXiv:1909.11114.

[7] B.Q. Huang *et al.*, "A New Feature Set with New Window Techniques for Customer Churn Prediction in land-Line Telecommunications," *Expert Systems with Applications*, vol. 37, no .5, 2010, pp. 3657–65.

[8] Y. Huang, B. Q. Huang, and M. T. Kechadi, "A New Filter Feature Selection Approach for Customer Churn Prediction in Telecommunications," *2010 IEEE Int'l. Conf. Industrial Engineering and Engineering Management*, Macao, 2010, pp. 338–42. DOI: 10.1109/IEEM.2010.5674306.

[9] I. Ullah *et al.*, "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector," *IEEE Access*, vol. 7, 2019, pp. 60,134–499. DOI: 10.1109/ACCESS.2019.291499.

[10] A.K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer Churn Prediction in Telecom Using Machine Learning in Big Data Platform," *J Big Data*, vol. 6, no. 28, 2019; https://doi.org/10.1186/s40537-019-0191-6

[11] A. De Caigny *et al.*, "Incorporating Textual Information in Customer Churn Prediction Models Based on A Convolutional Neural Network," *Int'l. J. Forecasting*, vol. 36, no. 4, 2020, pp. 1563–78.

[12] N. Alboukaey, A. Joukhadar, and N. Ghneim, "Dynamic Behavior Based Churn Prediction in Mobile Telecom," *Expert Systems with Applications*, vol. 162, 2020, p. 113,779.

[13] R. Prashanth, K. Deepak, and A. K. Meher, "High Accuracy Predictive Modelling for Customer Churn Prediction in Telecom Industry," P. Perner, Ed., *Machine Learning and Data Mining in Pattern Recognition*, MLDM 2017, LNCS, vol 10358, Springer 2017, pp. 309–402; https://doi.org/10.1007/978-3-319-62416-7_28.

[14] N. Bhargava *et al.*, "Decision Tree Analysis on J48 Algorithm for Data Mining," *Int'l. J. Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013, pp. 1114–19.

[15] C. G. Mena *et al.*, "Churn Prediction with Sequential Data and Deep Neural Networks. A Comparative Analysis," 2019, arXiv preprint arXiv:1909.11114.

Biographies

MUHAMMAD USMAN completed his Ph.D. in computer and information sciences at Auckland University of Technology, New Zealand. Currently, he is working at Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan, as an associate professor. He is also the principal investigator (PI) of the Predictive Analytics Lab, established under Pakistan's National Center of Big Data and Cloud Computing. His research interests include machine learning, data science, and predictive analytics.

WASEEM AHMAD received his M.Sc. in intelligent systems from Birmingham University, United Kingdom, in 2008 and his Ph.D. degree in artificial intelligence from Auckland University of Technology in 2012. From 2012 to 2015, he worked as a lecturer at Auckland University of Technology. In 2015, he moved to ToiOhomai Institute of Technology, New Zealand, where he has been working as an academic staff member. His research interests include nature inspired computing, deep learning, data visualization, genetic algorithms, and ensemble learning.

ALVIS FONG holds four degrees in electrical engineering and computer science. He was formerly a professor of computer science at Auckland University of Technology. He is now with Western Michigan University. His research interests include communications, intelligent systems, and multimedia. He has published 200 papers in these areas.