Humans Learning from Machines: Data Science Meets Network Management

Murugaraj Odiathevar*, Duncan Cameron*, Winston K.G. Seah, Marcus Frean, Alvin Valera
School of Engineering and Computer Science, Victoria University of Wellington
P.O. Box 600, Wellington 6140, New Zealand
{murugaraj.odiathevar, duncan.cameron, winston.seah, marcus.frean, alvin.valera}@ecs.vuw.ac.nz

***Abstract*—Internet Service Providers need to deploy and maintain many wireless sites in isolated or inaccessible terrain to provide Internet connectivity to rural communities. Addressing failures at such sites can be very expensive, both in identifying the fault, and also in the repair or rectification. Data monitoring can be useful, to spot anomalies and predict a fault (and possibly pre-empt it altogether), or to locate and isolate it quickly once it causes an issue for the network. There might be hundreds of variables to be monitored in principle, but only a few of significance for detecting faults. Here, in a case study involving a Wireless Internet Service Provider (WISP) in a rural area, we first illustrate a bottom-up approach to the identification of variables likely to be of use in an automatic anomaly detector. For the purpose of this study, the detector consists of an autoencoder neural network with weights optimized by machine learning (ML). We then show how the cause of an anomaly can be derived from indirect measurements, and use the model to learn relationships between certain variables.**

***Index Terms*—Machine Learning, Anomaly Detection, Feature Identification**

I. Introduction

Sparsely populated remote and rural communities make it tough to justify expensive – often financially unviable – fiber deployments, should they be installed overhead on shared power infrastructure, or buried in the ground. In order to deliver high-speed access services to such communities, many Internet Service Providers (ISPs) provide Internet connectivity through wireless means. Wireless deployments that serve such communities are often situated in difficult to reach locations, such as highlands, as these locations tend to provide the best coverage at the lowest cost. Consequently, grid-power is often unavailable, and renewable energy sources such as solar or wind must be used.

Regular maintenance of remote sites is expected of any quality ISP. However, there are occasions where unexpected failures occur, even when preventative maintenance are regularly conducted. Failure events, such as those caused by faulty batteries, defective solar controllers, or electromagnetic interference, can be difficult to accurately diagnose remotely. When physical access to sites is restricted, it is within the network managers best interest to monitor as many potential fault variables as possible.

Our work proposes a method to minimise the risks faced by network managers, through understanding the types of operational disruptions faced, and identifying what variables are important to monitor. In addition, we describe the procedure used to identify appropriate variables to monitor, so that a machine learning (ML) model can effectively detect any unexpected behaviour. Building from our approach, we are able to focus on, and identify, the cause of detected anomalies using feature identification methods.

Robust power systems are critical to any off-grid site remaining online every day of the year. Traditionally, to monitor the power of a site, a network manager might observe easily visualised variables, such as the battery bank voltage and charge current at a site. When the voltage of a battery bank falls below a recognised threshold, action needs to be taken. Generally, such monitoring systems are built based on the professional experience of network managers, rather than a recognised best practice [1], [2]. Consequently, faults are often identified reactively, as opposed to proactively, and sometimes after the window to prevent impending trouble has passed.

We propose monitoring many more variables, and using ML to gain insights about faults and patterns at the sites. With this knowledge, a system that allows quick identification of the causes of faults can be designed. If such a system is unable to identify a cause, we can also conclude with high confidence that this is a new cause of fault and eliminate all other possible reasons. This saves valuable time and effort on the network manager's part.

We provide a case study of how to identify those important variables and build a model. This model helps to find the unseen relationships between the variables. Using these correlations, we identify the cause of the faults. The contributions of this work are the following:

(i) A Bottom-Up approach to identify important variables for ML.
(ii) Using feature identification methods to identify the causes of anomalies.
(iii) An iterative procedure to update input variables.

The remainder of this paper is organised as follows: Section 2 surveys research efforts related to the management of WISPs, especially those in rural communities, using data science and ML approaches; Section 3 describes the methodology; Section 4 explains the experiment and the results, and we conclude the paper in Section 5.

*The first and second author contributed equally on different aspects of the study.

II. Related Work

The emergence of wireless mesh networks (around 2000) ignited interest in providing Internet connectivity to rural communities [2]. Since then, wireless networking technology has advanced significantly and prices have also become more affordable, yet a key obstacle preventing the adoption of advanced technology and use of better equipment in rural community networks is the low return on investment (ROI) [3]. Challenges faced by rural community network providers also include power management, difficulty in accessing network locations for maintenance and repairs, and unreliable links due to weak signals or interference, which further add to the operation costs. Consequently, much of the research in rural community networks has focused on addressing these issues [4]–[6].

In the early days of rural wireless mesh networks, dedicated hardware and software watchdogs were utilized to avoid having to make physical visits to remote and difficult access sites where network equipment were installed [2]. At each network node, an independent management module tracked the node and link(s) health and communicated the status information by Short Message Service (SMS) over a cellular network. This information was then used to diagnose network faults and decide whether a visit was necessary. As a large number of rural community mesh networks do not have cellular coverage, it is infeasible to implement a cellular-based control channel. Instead, another system's network management component used the backhaul link via the gateway to relay any control messages [7]. It also logged essential information for network diagnosis and post-failure analysis.

Networking and distributed computing have long been the enablers of other technologies, and undoubtedly been providing efficient computation resources as well as data acquisition capabilities for ML and big data applications. The increased complexity of networks and the traffic carried by them also motivated the application of ML and big data techniques for both mundane and complex problems encountered in network operation and management [8]–[10]. Due to their significantly larger customer base, higher revenue and ROI, cellular networks [11]–[15] receive much more interest than community networks [10] in the use of ML and big data. Another clear orthogonal trend is the application of ML for network traffic classification [15]–[17] and anomaly detection to address security issues [10], [18], [19] that can be applied to all forms of networks.

Boutaba *et al.* [9] provide a comprehensive survey on ML for networking, of which the fault management aspect is of particular interest to us. They stated that fault management "requires network operators to have a thorough knowledge of the entire network, its devices and all the applications running in the network." Most critical network infrastructures come with uninterrupted power and operators focus on dealing with network-related faults. However, power management remains a key challenge faced by WISPs for rural communities as the reliance on batteries [4], [20] or off-grid renewable power sources (e.g. solar or wind) [21] is inevitable. Not only does loss of power disrupt network operations, low-quality power damages networking equipment as well as batteries [2]. Power management is therefore a critical aspect of rural community WISPs and forms an integral component of the entire network together with the other network devices. This consideration has been adopted in the context of green networks [22].

ML has also been applied in the management of power systems [23], [24] and grids [25], as well as renewable power systems [26], [27] but, to the best of our knowledge, not in the context of power management within network operation and management of rural community networks by WISPs.

III. Methodology

Each ISP is unique and operates under different requirements and scenarios. ISPs may already subscribe to a third party data monitoring tool such as SolarWinds [28]. An ISP can also set up a monitoring tool from open source programs such as Zabbix [29]. In the former, an ISP has an abundant number of variables that can be monitored, and in the latter, an ISP would need to build the variables up from scratch.

Even with an abundant number of variables, it is futile to indiscriminately throw all the variables into a ML algorithm in hope of learning some patterns. Not only will it be difficult, but patterns learnt cannot be easily interpreted. In the literature, there are many feature selection methods that reduce a set of variables to important ones [30]. We shall refer to this approach as the *Top-Down* approach. For this method, we need real world data that depicts anomalies. Also, it may not provide enough context or meaning to gain proper insights into the faults. Thus, it is prudent to build the monitoring variables from scratch; in other words, from the knowledge of the network manager or the *Bottom-Up* approach. The Bottom-Up approach also allows managers to incorporate new variables that may be important but are not in the list of variables provided by third party monitoring solution providers, for example installing a sensor to monitor moisture or temperature near the batteries. There is also a fine balance between attempting to monitor the variables and being able to measure them accurately.

A. Bottom-Up Approach

Being aware of what variables are fed into the ML model is an important step which is usually ignored. We input many variables and expect it to perform. But if we put in uninformative variables, we will not obtain any useful results. Informative variables should come from our knowledge of the system. We must help the ML model so that it can help us. Here, we introduce the Bottom-Up approach to identify these variables. To begin, we asked the network manager the following questions.

1) Describe the nature of the faults or anomalies faced by the system.
2) To detect each of the above mentioned faults, which variables are currently monitored?

3) For each of the above mentioned variables, how are the data collected?

The rationale for the first question is to understand the faults that occur within the site. This can be battery faults, latency issues, radio frequency anomalies, etc. Each of these can be further broken down into sub-categories. For instance, battery faults can be caused by either low-voltage, battery out of order or limited solar generation. We can bring it all the way to the individual battery or each link in the network. The objective is to make this list as fine-grained and as comprehensive as possible.

Next, from each of the sub-categories, we can identify variables which the network manager currently uses to detect it. For example, high *Watt hours* and high *Current* can result in low voltage situations which affects performance as a whole. One of the difficulties in identifying these variables for ML is that network managers tend to understand the faults from a high level. "A low-voltage situation could be caused by excessive consumption or a lack of control of power generation." For ML methods, this cause needs to be further fine-tuned to specific measurable variables by asking appropriate questions. Some examples include "How to measure consumption or power-generation?" It also helps the network manager think through their current processes and facilitates the building of a model of the fault that the data can point to when it occurs.

B. Measurement

The next challenge is to accurately measure the identified variables. If the existing third party provider already measures and provides all of the required variables, then the task is simple. Else, one would need to find a way to measure the variables directly or indirectly depending on the site architecture and set-up. For example, the 'ping' command can be used to measure round-trip-time (RTT), weather information can be scraped off the web, and signal-to-noise ratio (SNR) measurements – as well as other spectrum analysis – can be performed using existing radios, or by setting up custom monitoring hardware within the environment.

Another challenge is to determine the appropriate time-window to make the measurements. This depends on the persistence of the site operation when a fault is detected. It also depends on the measuring capabilities. Some variables can only be measured every hour, while others can be measured every few seconds. Small intervals capture fine changes which may not be relevant in the context of site operation. It may also require more computation to measure, and additional storage space. Rendering an interval too large might miss pertinent information. Either way, it is necessary to study different time-windows during experimentation to gain specific insights.

C. Types of Variables

Certain variables are not numerical. For example the weather is a categorical variable and certain device information such as *state* of the device, *flags*, or *DIP switches* which only take a few numerical values should also be considered as categorical. Some faults are detectable relative to previous values. This goes into second order calculation of variables such as *Change in Watt Hours* or 'Δ Watt Hrs' over two time-windows.

In a time-window, certain variables can be measured many times. For instance, *Battery Temperature* or *Charge Current* may be measured every 10 seconds. In a time-window of 5 minutes, one obtains 30 values. Statistical attributes such as mean, maximum, minimum, median, variance, and interquartile-range of variables can be used to describe the distribution of the 30 values. This depends on the variable and the anomaly of interest, i.e., which statistical attribute indicates the fault most accurately and is based on the knowledge and understanding of the network.

IV. Experiment with ISP

In this section, we will describe the engagement with an ISP to utilise ML methods to gain insights.

A. ISP Details

We worked with Venture Networks, a rural ISP located in the Horowhenua District of New Zealand. The ISP provides internet connectivity to local farms, and has several sites across the region. Some of the sites are difficult to access, especially during the winter months, due to the challenging terrain and severe weather. In the last year, Venture Networks has been caught off guard by several unexpected failures at their sites. In one case, a battery bank failed well before expected, and in another case, water penetrated a supposedly waterproof power junction box, causing half of the solar array to fail.

Network managers currently have the ability to monitor important variables using a variety of free and proprietary software. Faults are generally found based on conditional logic, and network managers are typically alerted by email, and sometimes over mobile push notifications or SMS. Upon notification of a fault, manual work is typically required to identify the root cause. When dealing with sites in challenging areas, often a trip to identify the fault will be required, which can be expensive, and worse, extremely dangerous especially during adverse weather conditions.

B. Variables Identified

The ISP subscribes to a monitoring software provider that provides monitoring of over 100 different variables. Through engagement and the processes mentioned in Section III-A, we identified the following variables depicted in Table I which are related to three main types of anomalies that have occurred frequently. These variables are a natural starting point to measure and input into ML methods. Through analysis and identification of faults, more variables can be identified and added later on as anomalies occur. For example, if a hardware fault is found to be caused by water damage, that cause could be incorporated by measuring moisture or water levels in the vicinity as a variable for future detection.

Not all of the variables are provided by the ISP. Variables such as the weather were recorded by polling a weather service Application Programming Interface (API), due to the lack

TABLE I
MEASURED & DERIVED VARIABLES

Variables	Description	Type	Derived Variables	Type
Battery Voltage	Instantaneous battery bank voltage (internal sensor)	Numerical	ΔBattery Voltage	2^{nd} Order
Battery SVoltage	Instantaneous battery bank voltage (external sensor)	Numerical	ΔBattery SVoltage	2^{nd} Order
Target Voltage	Target battery bank voltage	Numerical	–	–
Charge Current	Instantaneous charge controller charge current	Numerical	–	–
Output Power	Instantaneous power into the battery bank	Numerical	ΔOutput Power	2^{nd} Order
Input Power	Charge controller input power	Numerical	ΔInput Power	2^{nd} Order
Array Voltage	Instantaneous output voltage of solar array	Numerical	ΔArray Voltage	2^{nd} Order
Array Current	Instantaneous output current of solar array	Numerical	ΔArray Current	2^{nd} Order
Sweep Vmp	Maximum power voltage of the array	Numerical	–	–
Sweep Voc	Open circuit voltage of the array	Numerical	–	–
Sweep Pmax	Maximum power produced by the array	Numerical	–	–
Battery Temp	Battery temperature (external sensor)	Numerical	ΔBattery Temp	2^{nd} Order
Heat sink	Charge controller heat sink temp	Numerical	ΔHeat sink	2^{nd} Order
Amp Hours	Daily (moving) amp hours count	Numerical	–	–
Watt Hours	Daily (moving) watt hours count	Numerical	–	–
Weather Temp	Levin town temperature	Numerical	–	–
Weather Wind	Levin town wind speed	Numerical	–	–
Tx Capacity	Transmit capacity of the wireless link	Numerical	–	–
Rx Capacity	Receive capacity of the wireless link	Numerical	–	–
Signal	Received Signal Strength Indicator	Numerical	–	–
Noise Floor	Noise floor of the wireless link	Numerical	–	–
SNR	Signal-to-Noise Ratio of the wireless link	Numerical	–	–
RTT	Site-to-Site Round-trip Time	Numerical	–	–
Tx RTT	Radio transmit Round-trip Time	Numerical	–	–
Min Vb daily	Daily (moving) min battery voltage	Statistical	–	–
Max Vb daily	Daily (moving) max battery voltage	Statistical	–	–
Min Tb daily	Daily (moving) min battery temp	Statistical	–	–
Max Tb daily	Daily (moving) max battery temp	Statistical	–	–
Weather	Weather state (clouds/rain/sun, etc)	Categorical	–	–
Charge State	Current stage of the 4-stage charging algorithm	Categorical	–	–
DIP Switches	Hardware configuration switch state	Categorical	–	–
CCQ	Client Connection Quality	Categorical	–	–

of a local weather station. Most of the variables – except those related to weather, latency, or radio-specific functions – were captured using a commercially available solar charge controller. A time window of 5 minutes was used to capture each of the variables, and the results were stored on a server running within Venture Networks' central office. While we had access to a charge controller that supported normal TCP/IP communications, we acknowledge that not all charge controllers have remote monitoring and configuration capabilities, which is a future limitation that we will need to overcome.

C. Preprocessing

ML models require certain preprocessing to learn effectively. They are as follows.

- Categorical variables need to be converted to binary or numerical values. We used one-hot encoding to convert the categorical variables to binary and Principal Component Analysis (PCA) without any dimensionality reduction to map it to the eigenspace. For more details on these methods, we refer readers to [31], [32].
- The data should be normalised to a range between [0,1] or [-1,1]. This is important because the large values in certain variables would put higher weight and importance on those variables and reduce the importance of other variables. We used Min-Max scaling [33] to accomplish this.

D. Model

Note that chronological ordering of the data is a requirement for some modelling methods such as time-series. However, it requires the assumption that the data in future time periods depend on the past time periods and this is not always true.

For this specific ISP, the data profile significantly varies during different hours of the day. Training one model over 24 hours would not suffice because if a situation that would usually occur during the night occurred during the day, that would be regarded as unexpected. We insert a new categorical variable to denote the hour interval.

This study is not about showing which is a better ML method but how to use ML in daily operations. Thus, we will not be comparing different methods. We choose the AutoEncoder (AE) because it has been used in many studies on anomaly detection [34]. An AE consists of an encoder network, a latent layer and a decoder network as shown in Figure 1. The encoder maps the input data into the latent layer of lower dimensionality, and the decoder reconstructs them again. This is also called an undercomplete AE. The weights of the maps are determined using the data during the training phase. Training an undercomplete AE forces the AE to keep important variations and to find the complex relationships between the variables that are required to reconstruct the input. They also provide a Reconstruction Error (RE) which can be used as an anomaly score [34]. The AE is trained with only normal data while minimising RE. Thus, the assumption is that normal data can be reconstructed while anomalous data will have higher RE.

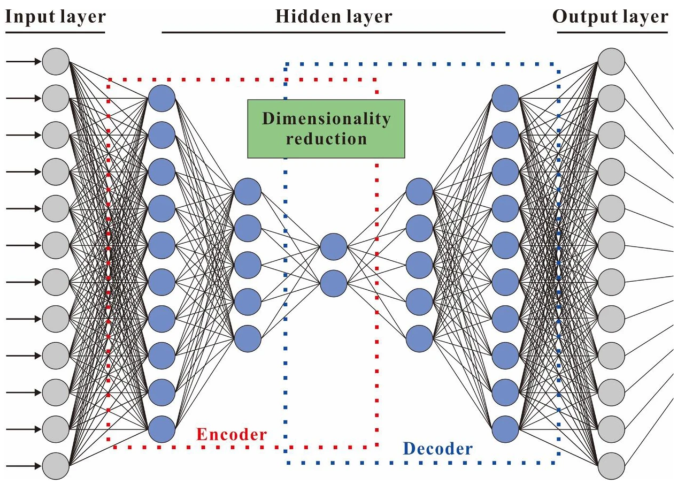

Fig. 1. AE Model [35]

For the training, we use data that the network administrator has identified as normal day operations. We use an AE with five hidden layers with the number of neurons in the middle layer equalling $1 + \sqrt{d}$, where d is the number of input variables [36], [37]. We used batch training with the leaky Relu activation function at each layer and Sigmoid on the last layer.

To show the ability of ML to identify anomalies based on other variables, we removed Battery Voltage, Battery SVoltage, ΔBattery Voltage and ΔBattery SVoltage variables from training and testing.

E. Results

We assessed the model's performance over a few days where there were some unexpected battery behaviour. Figure 2 shows how Battery Voltage changes during the course of the day. During normal operations, the anomaly scores are low. However, when there is an unexpected behaviour there are more spikes in anomaly scores. Note that the AE has determined this solely based on other variables. This suggests that if Battery Voltage were not observed, the AE would still be able to detect the unexpected behaviour. Upon further analysis, the network manager can determine that Battery Voltage should be measured and be an input variable to the AE to make it more robust. The bottom graph of Figure 2 shows how one variable moves. The top graph shows how all of the other variables move in correlation depicted on to one dimension to show whether it is moving normally or anomalously. This demonstrates the power of a ML model and the benefits of using it to uncover hidden relationships.

From the network manager's method of observing 'Battery Voltage', the problem was detected on the 17th of June after consecutive days of torrential rain. The model also detects more anomalies during that period and after. This is

TABLE II
FEATURES THAT DESCRIBE UNEXPECTED BEHAVIOUR IN FIGURE 2

	Ranking based on	
Rank	χ^2 **statistic**	**Mutual Information**
1	Sweep Voc	Sweep Voc
2	Sweep Vmp	Min Vb daily
3	Array Voltage	Watt hours
4	Target Voltage	Amp hours
5	Charge Current	Max VB daily
6	Output Power	Max TB daily
7	Input Power	Weather Temp

determined by comparing the anomaly scores during normal operations on left top of Figure 2. The threshold in blue is also determined in this manner. A few data points being above the threshold during normal operations is not surprising and could reflect measurement error. However, when many points are above the threshold, we can conclude that there is some unexpected behaviour.

To determine the cause of the unexpected behaviour, we took the data during the anomalous period and performed feature identification. We ranked the features based on χ^2 statistic [38] and Mutual Information (MI) [39] in TABLE II. Both these methods make different assumptions about the data but regardless, it helps us to identify that the unexpected behaviour is due to the battery. Note that in the literature, these methods are used for feature identification before training a ML model [30]. We do this ***after*** identifying the anomaly to understand what type it is. This also helps us identify other important variables to monitor and the hidden relationships between them in describing the fault. For instance, we should also monitor 'Sweep Voc' instead of only 'Battery Voltage'. We also learn about the other related variables to this battery anomaly. The more variables we can measure and input into the AE, the more powerful and effective this analysis will be.

We also artificially injected latency anomalies by using the Linux utility traffic control (tc), which ran on the server collecting anomaly data, so as to not degrade live customer connections. The results are shown in Figure 3. The results are shown in logarithmic scale since the anomaly scores were huge. The respective features are ranked in TABLE III. However, by looking at the contribution of each feature to the anomaly score based on the values of the χ^2 statistic and Mutual Information (MI), only RTT stood out. This is because the latency anomaly was injected artificially instead of it having occurred in real-time. Despite that, we learn that unexpected values in 'Input Power', 'Array Voltage' and 'Sweep Voc' can also be related to latency anomalies.

V. CONCLUSION

Hence from the above results, the ML model is able to detect unexpected behaviour. We must be careful in providing it with the right variables. A hunch that certain variables can help to detect anomalies can be a good starting point. This is the bottom-up approach described in section III-A. Subsequently, as anomalies are detected, feature identification

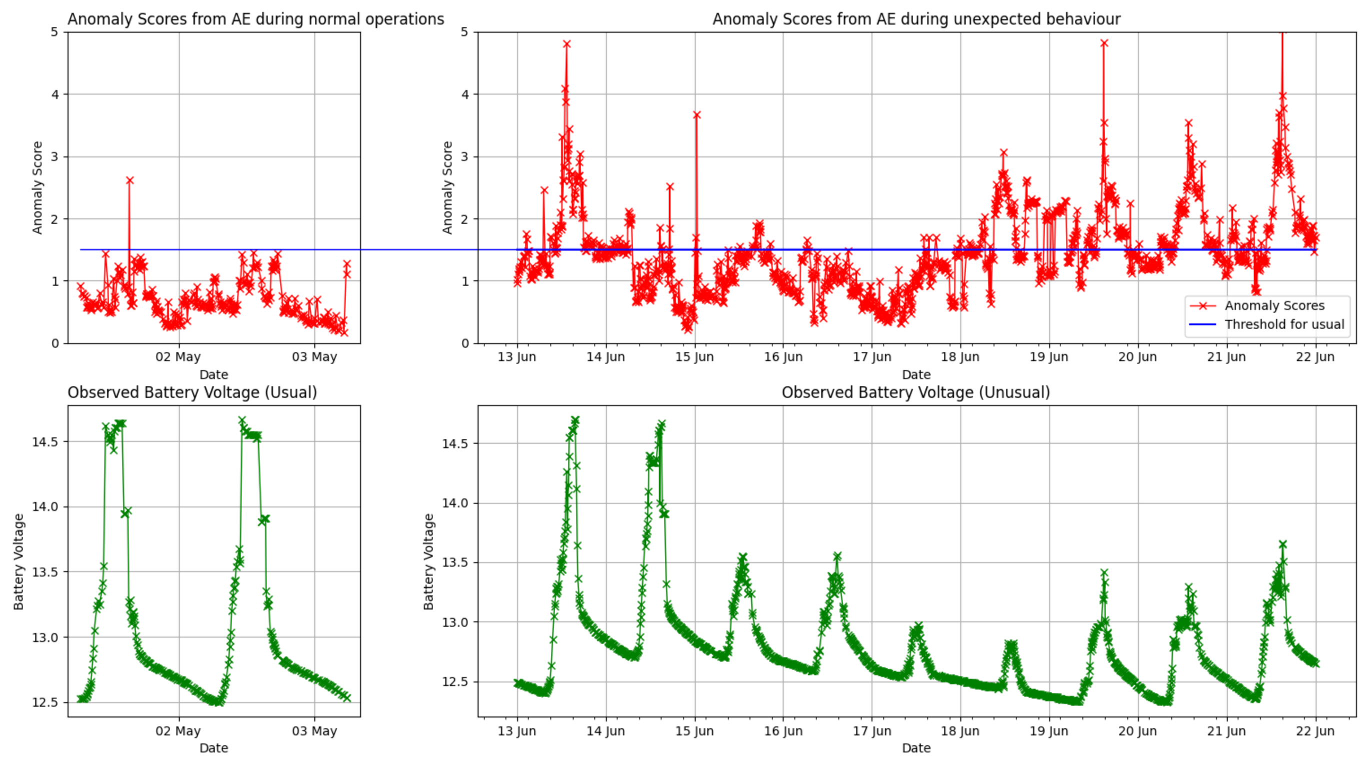

Fig. 2. Left Bottom: Observed Battery Voltage variable during normal operations. Left Top: Anomaly scores from AE based on other variables during normal operations. Right Bottom: Observed Battery Voltage during some unexpected behaviour. Right Top: Anomaly scores based on other variables during unexpected behaviour.The threshold in blue is defined based on Left Top graph.

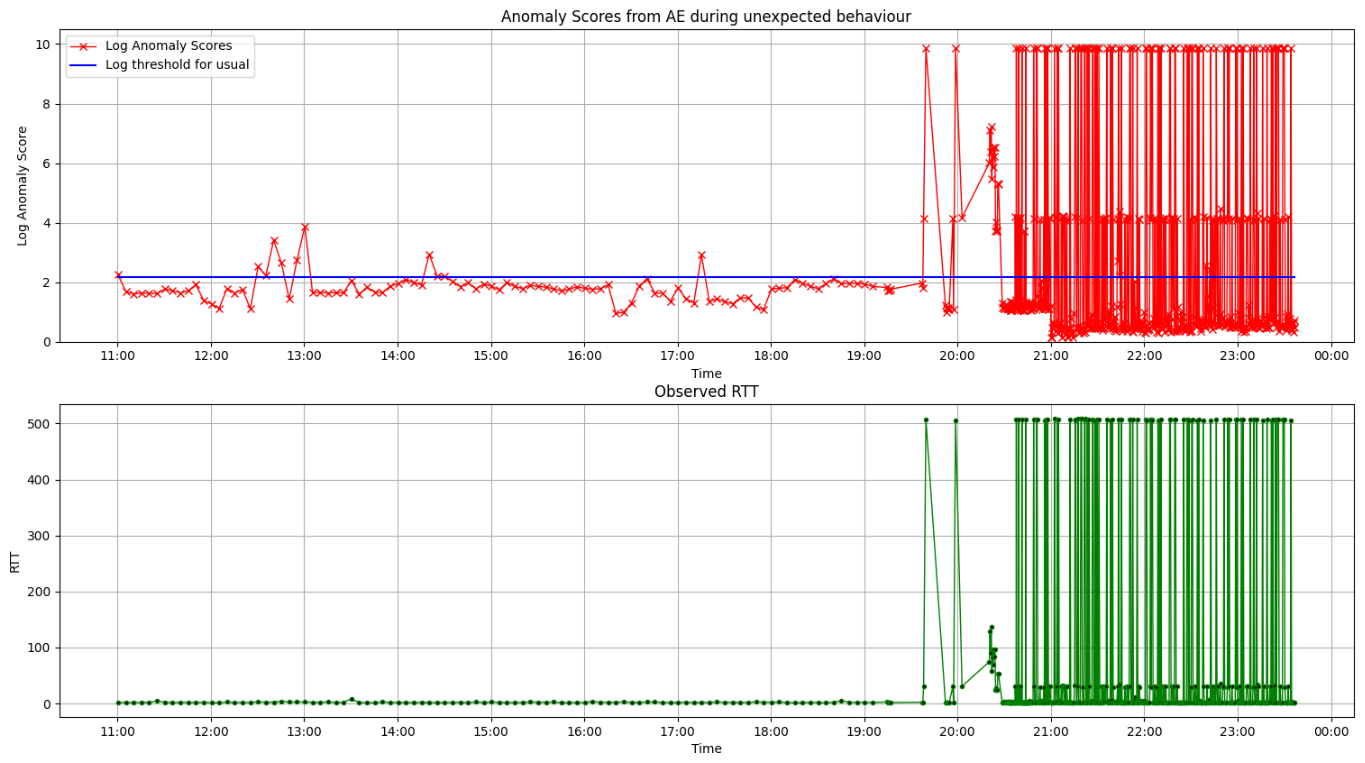

Fig. 3. Top: Log anomaly scores from AE based on other variables during normal operations and during injected latency after 1930. Bottom: RTT. The threshold in blue is the same value as in Figure 2. All log anomaly scores are translated to above 0.

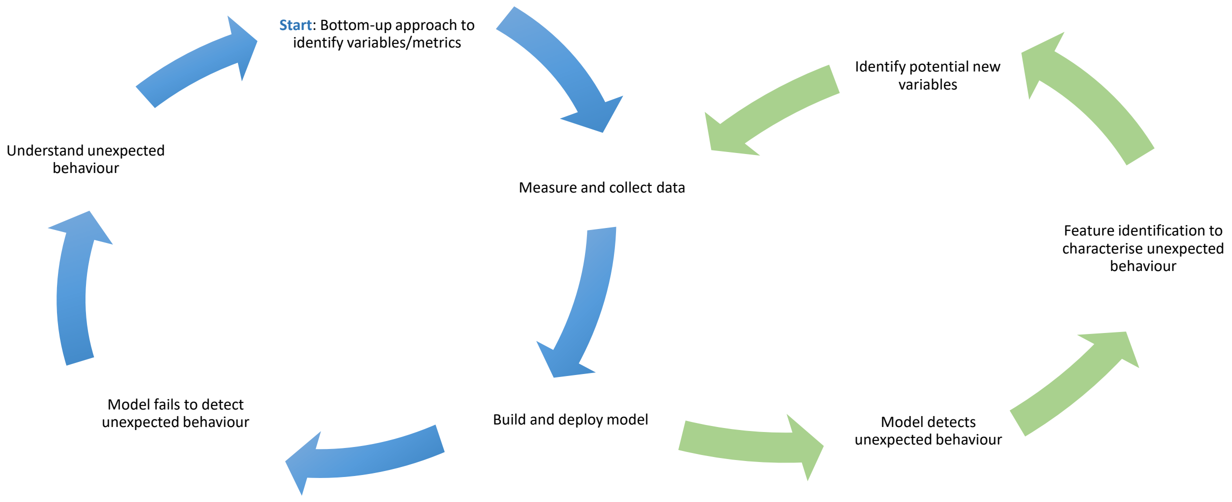
Start: Bottom-up approach to identify variables/metrics
Measure and collect data
Build and deploy model
Model fails to detect unexpected behaviour
Understand unexpected behaviour
Model detects unexpected behaviour
Feature identification to characterise unexpected behaviour
Identify potential new variables

Fig. 4. An iterative procedure to identify variables and build the model. After a couple of iterations, we will reach the green loop where the model detects all related anomalies for the network manager.

TABLE III
FEATURES THAT DESCRIBE LATENCY IN FIGURE 3

	Ranking based on	
Rank	χ^2 **statistic**	**Mutual Information**
1	RTT	RTT
2	Target Voltage	Sweep Voc
3	Sweep Vmp	∆Input Power
4	Array Voltage	Array Voltage
5	Sweep Voc	Input Power
6	Output Power	Weather Temp
7	Input Power	Sweep Pmax

can be done to categorise the fault and to find important relationships between the variables that describe the fault. The network manager can identify potential new variables to measure for fine grained results in future for this specific unexpected behaviour. If an anomalous behaviour is missed, which the network manager has been made aware of from other sources such as customer complaints, it means that the ML model has not been provided with the right variables to detect that particular anomaly. The next time that particular anomaly occurs, it can be easily detected and mitigated. Hence, this is an iterative process depicted in Figure 4 to identify variables over time and continually build a robust model.

This process also helps the network manager understand exactly how the network operates, identify the specific cause(s) of a fault and address it instead of using ‘patchwork’ methods such as boosting the signal or increasing power. After a robust model is built, this also allows for automation of network operations which will be addressed as future work of this research.

VI. ACKNOWLEDGEMENTS AND CONTRIBUTIONS

Murugaraj Odiathevar’s contributions come from the data science perspective. This includes the method for identifying the variables, performing preprocessing, training, anomaly detection and analysis. His work is supported by a Wellington Doctoral Scholarship awarded by Victoria University of Wellington.

Duncan Cameron manages Venture Networks, and brings the ISP perspective. His contributions include identifying the variables, measuring the data, implementation and verification.

Finally, the authors would like to acknowledge Susanne Cameron for taking the time to proofread the manuscript.

REFERENCES

[1] A. Nungu, R. Olsson, and B. Pehrson, “On powering communication networks in developing regions,” in *Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC)*, June 2011, pp. 383–390.

[2] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer, “Beyond pilots: Keeping rural wireless networks alive,” in *Proceedings of the 5th USENIX Symposium on Networked Systems Design & Implementation (NSDI 2008)*, San Francisco, CA, USA, 16-18 April 2008.

[3] S. M. Mishra, J. Hwang, D. Filippini, R. Moazzami, L. Subramanian, and T. Du, “Economic analysis of networking technologies for rural developing regions,” in *Internet and Network Economics*, X. Deng and Y. Ye, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 184–194.

[4] D. R. Paudel, K. Sato, B. P. Gautam, and D. Shrestha, “Design and implementation of partial mesh community wireless network in Himalayan region,” in *Proceedings of the Third Asian Himalayas International Conference on Internet (AHICI)*, Kathmandu, Nepal, 23-25 Nov 2012, pp. 1–6.

[5] C. Rey-Moreno, Z. Roro, W. D. Tucker, M. J. Siya, N. J. Bidwell, and J. Simo-Reigadas, “Experiences, challenges and lessons from rolling out a rural wifi mesh network,” in *Proceedings of the 3rd ACM Symposium on Computing for Development*, ser. ACM DEV ’13. New York, NY, USA: ACM, 2013, pp. 11:1–11:10. [Online]. Available: http://doi.acm.org/10.1145/2442882.2442897

[6] S. Hasan, Y. Ben-David, M. Bittman, and B. Raghavan, “The challenges of scaling wisps,” in *Proceedings of the 2015 Annual Symposium on Computing for Development (DEV ’15)*, London, UK, Dec 2015.

[7] V. Gabale, R. Mehta, J. Patani, K. Ramakrishnan, and B. Raman, “Deployments Made Easy: Essentials of Managing a (Rural) Wireless Mesh Network,” in *Proceedings of the 3rd ACM Symposium on Computing for Development*, Bangalore, India, Jan 2013.

[8] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, March 2018.

[9] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.

[10] L. Maccari and A. Passerini, "A big data and machine learning approach for network monitoring and security," *Security and Privacy*, vol. 2, no. 1, p. e53, 2019.

[11] P. Casas, P. Fiadino, and A. D'Alconzo, "Machine-Learning Based Approaches for Anomaly Detection and Classification in Cellular Networks," in *Proceedings of the IFIP Traffic Monitoring and Analysis workshop (TMA)*, Louvain La Neuve, Belgium, 7-8 April 2016.

[12] P. Casas, "Machine learning models for wireless network monitoring and analysis," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Barcelona, Spain, 15-18 April 2018, pp. 242–247.

[13] L. Pan, J. Zhang, P. P. Lee, M. Kalander, J. Ye, and P. Wang, "Proactive microwave link anomaly detection in cellular data networks," *Computer Networks*, vol. 167, p. 106969, 2020.

[14] B. Li, S. Zhao, R. Zhang, Q. Shi, and K. Yang, "Anomaly detection for cellular networks using big data analytics," *IET Communications*, vol. 13, no. 20, pp. 3351–3359, 2019.

[15] I. Hadj-Kacem, S. B. Jemaa, S. Allio, and Y. B. Slimen, "Anomaly prediction in mobile networks : A data driven approach for machine learning algorithm selection," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary, 20-24 April 2020, pp. 1–7.

[16] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[17] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.

[18] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 70–73, Apr. 2014.

[19] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[20] A. Hameed, A. N. Mian, and J. Qadir, "Low-cost sustainable wireless internet service for rural areas," *Wireless Networks*, vol. 24, pp. 1439–1450, 2018.

[21] D. Cameron, A. Valera, and W. K. G. Seah, "Elasticwisp: Energy-proportional wisp networks," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary, 20-24 April 2020, pp. 1–9.

[22] L. X. Cai, H. V. Poor, Y. Liu, T. H. Luan, X. Shen, and J. W. Mark, "Dimensioning network deployment and resource management in green mesh networks," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 58–65, 2011.

[23] N. Hatziargyriou, *Machine Learning Applications to Power Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 308–317.

[24] H. A. Tokel, R. A. Halaseh, G. Alirezaei, and R. Mathar, "A new approach for machine learning-based fault detection and classification in power systems," in *Proceedings of the IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, 19-22 Feb 2018, pp. 1–5.

[25] R. Eskandarpour and A. Khodaei, "Machine learning based power grid outage prediction in response to extreme events," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3315–3316, 2017.

[26] K. S. Perera, Z. Aung, and W. L. Woon, "Machine learning techniques for supporting renewable energy generation and integration: A survey," in *Proceedings of the International Workshop on Data Analytics for Renewable Energy Integration (DARE)*. Dublin, Ireland: Springer International Publishing, 10 Sept 2014, pp. 81–96.

[27] S. Theocharides, G. Makrides, G. E. Georghiou, and A. Kyprianou, "Machine learning algorithms for photovoltaic system power output prediction," in *Proceedings of the IEEE International Energy Conference (ENERGYCON)*, Limassol, Cyprus, 3-7 June 2018, pp. 1–6.

[28] "Solar Winds." [Online]. Available: https://www.solarwinds.com/

[29] R. Olups, *Zabbix 1.8 network monitoring*. Packt Publishing Ltd, 2010.

[30] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[31] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[32] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics) springer-verlag new york," *Inc. Secaucus, NJ, USA*, 2006.

[33] J. Grus, *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[34] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48 231–48 246, 2018.

[35] K. Nam and F. Wang, "The performance of using an autoencoder for prediction and susceptibility assessment of landslides: A case study on landslides triggered by the 2018 Hokkaido Eastern Iburi earthquake in Japan," *Geoenvironmental Disasters*, vol. 6, no. 1, p. 19, 2019.

[36] M. Nicolau, J. McDermott *et al.*, "A hybrid autoencoder and density estimation model for anomaly detection," in *Int'l. Conf. on Parallel Problem Solving from Nature*. Springer, 2016, pp. 717–726.

[37] V. L. Cao, M. Nicolau, and J. McDermott, "Learning Neural Representations for Network Anomaly Detection," *IEEE Trans. on Cybernetics*, vol. 49, no. 8, pp. 3074–3087, Aug 2019.

[38] D. P. Vivencio, E. R. Hruschka, M. do Carmo Nicoletti, E. B. dos Santos, and S. D. Galvao, "Feature-weighted k-nearest neighbor classifier," in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symp. on*. IEEE, 2007, pp. 481–486.

[39] W. Duch, *Filter Methods*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 89–117.