Detecting malicious signal manipulation in smart grids using intelligent analysis of contextual data.

MAJDANI, F., BATIK, L., PETROVSKI, A. and PETROVSKI, S.

2020

Detecting Malicious Signal Manipulation in Smart Grids Using Intelligent Analysis of Contextual Data

Farzan Majdani, Lynne Batik, Andrei Ptrovski
School of Computing Science
Robert Gordon University
Aberdeen, United Kingdom
(f.majdani-shabestari, l.batik, a.petrovski)@rgu.ac.uk

Sergei Petrovski
School of Electric Stations
Samara State Technical University
Samara, Russian Federation
petrovski.sv@samgtu.ru

***Abstract*—This paper looks at potential vulnerabilities of the Smart Grid energy infrastructure to data injection cyber-attacks and the means of addressing these vulnerabilities through intelligent data analysis. Efforts are being made by multiple groups to provide to defence-in-depth to Smart Grid systems by developing attack detection algorithms utilising artificial neural networks that evaluate data communication between system components. The first priority of such algorithms is the detection of anomalous commands or data states; however, anomalous data states may also result from physical situations legitimately encountered by equipment. This work aims at not only detecting and alerting on anomalies, but at intelligent learning of the system behaviour to distinguish between malicious interference and anomalous system states occurring due to maintenance activity or natural phenomena, such as for instance a nearby lightning strike causing a short-circuit fault.**

***Index Terms*—Intelligent Analysis, Contextual Data, Artificial Neural Networks, Malicious Interference, Machine Learning, Smart Grid, SCADA Cybersecurity**

I. Introduction

Electrical infrastructures around the world are currently evolving from centralised, large-scale systems with components which are largely offline or on private networks, controlled where necessary by proprietary code, to much more complex distributed systems incorporating a wide range of “smart” software-controlled components and wider network connectivity. The benefits of this are many: these systems are able to incorporate multiple smaller power generators from commercial wind farms to individual consumer solar panels, and are able to route supply to demand in a far more efficient fashion [14]. The need for efficiency of management and operation of the increasingly complex landscape of technical components in order to limit costs also now means that far more internetworked software control systems are in play, and that many, if not most, components can be controlled as well as monitored remotely. However, this widens an attack surface for malicious interference from actors and groups which do not need geographic proximity to the systems they are attacking [14]. Given how much damage can be done at a large scale by taking electrical infrastructure offline, this is also an obvious target for state-funded and well-organised groups. It is in this context that defence-in-depth of Supervisory Control And Data Acquisition (SCADA) systems and Industrial Control Systems (ICS) is now a major concern. Traditionally, SCADA systems have had a long life cycle with few updates, but had limited vulnerability due to their relative isolation from the outside world. Now, with the greater technical complexity and connectivity, solutions far beyond patching are vital to protect from malicious interference. In recent years, part of the development of that defence-in-depth has been the addition of machine learning algorithms to monitor network traffic of both commands and sensor data, to detect both injected commands and injected data which may be used to indirectly manipulate the physical systems which are supposed to protect electrical components from damage [15]. This paper is concerned with the injection of false sensor data into an electrical substation’s network to simulate voltage under-load or overload, with the intention of making equipment tasked with maintaining voltage equilibrium respond inappropriately, thus creating an actual disequilibrium and destabilising the resource. What complicates detection of this type of attack, however, is that legitimate physical conditions, such as a lightning strike on or near the substation, or electrical transmission lines taken down for emergency maintenance, may result in sensor data almost indistinguishable from maliciously injected anomalous sensor readings. The specific focus of this research has been the potential use of external data along with machine learning profiling of “normal operation” vs. anomalous states in order to better distinguish malicious data injection from legitimate sensor states. The paper is structured as follows. In Section 2 we provide a literature review of some work on managing Smart Grids through detection of malicious signal manipulations. Section 3 covers the methodological aspect of the work we have carried out, including the source of the simulated data and novel data pre-processing procedures. Section 4 covers the results and issues of implementation and testing in more depth. Finally, Section 5 discusses the implications and further concerns, as well as steps which could take this research further.

II. Literature Review

A. Smart Grid technologies

In the past the energy grid has consisted of a large central power generation unit or a small number of central units, feeding high voltage current into a one-way high-voltage transmission and distribution network. This was stepped down to a lower voltage at substations geographically close to customer endpoints, and much of the automated equipment managing transmission and distribution was offline or had limited connectivity to central SCADA systems [12]. This system is being replaced by the far more decentralised Smart Grid, which not only includes traditional bulk power generators, but also ties in numerous smaller systems of generation including medium- and small-scale renewables. These may contribute energy at high, medium or low voltages, and they must be brought safely into the same transmission and distribution system for redistribution to consumers.

Decentralised control for safety and efficiency becomes a necessity in this grid architecture. The grid is often already capable of ramping power generation up or down according to variable demand (there are multiple load-following bulk generators across the UK), but the time-variant nature of energy distribution becomes vastly more complex with the addition of an inevitably variable supply of wind and solar, in which the level of generation is often mismatched to the level of demand. As parenthetically noted in [21], many traditional systems were designed to deal with "stable power systems with [a] fixed topology"; the changing, unstable topology of modern Distributed Energy Resources (DER) requires the integration of Intelligent Electronic Devices (IEDs) and ICT capabilities into sensors, voltage control, Remote Terminal Units (RTUs) and other components for grid stability and resilience [10]. These provide both real-time individual component and aggregate monitoring of the system load, and are able to efficiently manage stability of supply matched to endpoint demand as well as maintaining the flow of voltage and current through the grid components within acceptable operational limits. In addition, costs to the energy companies are reduced by allowing more manual control by remote operators, rather than technicians on-site. The energy grid is becoming a true cyber-physical system: demand-response and supply reliability are both becoming governed by computerised control systems over network communication, as noted in [5] and [16]. Unfortunately this creates a built-in vulnerability to remote malicious interference.

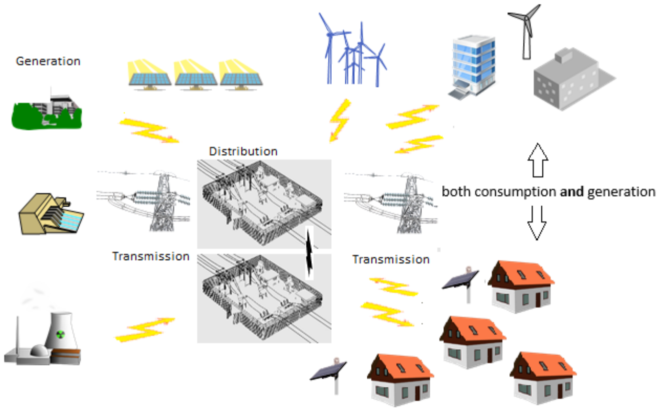

Fig. 1. Smart Grid Architecture

There are multiple possible forms of attack. Malware may penetrate a SCADA system from a connected central corporate or desktop computer, and then pivot to an Industrial Control Systems (ICS), or malware exploiting substation communication protocols governed by IEC 61850 may be injected directly into RTUs or other control units via their network connectivity [12]. Denial of service attacks may be used to shut down communication between DER components, or time-delay attacks may use intrusion or network flooding to delay time-critical commands or sensor data during transmission between components [20]. Commands or sensor data from legitimate sources may be intercepted on the network and either destroyed or altered; or "spoofed" commands or data originating from malicious actors may be fed into ICS networks with the appearance of legitimacy. These potential attack routes depend on targeting communication, sensing, or monitoring. There is an additional route of attack, not considered in depth here, of direct manipulation of a physical signal into a sensing device. For example, Ju [9] looked at how manipulation of reported bus voltage could be used to create massively inappropriate power injection into a DER which uses distributed, local control of reactive-power injection ("Volt/VAR control") to keep a resource at a voltage equilibrium, destabilising not just the resource but "the voltage profile across the distribution network" and resulting in widespread grid damage.

B. *Intrusion Detection Systems for Smart Grid*

Efforts to make the new Smart Grids more secure have been underway at every stage of Smart Grid development, but for the purposes of this research we are focusing on work done in this domain since 2014. One of the early papers looking at cyber attack on the grid through the manipulation of sensor data is Isozaki et al. [7]. This is worth consideration as it lays out quite clearly how the scenario for attack works, and a proposed defence.

The authors of [4] provide the energy datasets that are also used in a slightly altered form for our research. This work addresses a more complex situation than the one outlined above, and explicitly does so in a way which involves the Phasor Measurement Unit (PMU) which is an integral part of the cyber layer in the cyber-physical system that is the Smart Grid. The PMU collects and monitors the status of a wide array of system devices, "including relays, breakers, switches and transformers", in order to maintain a fine-grained control with low latency and very short reaction times. This paper also deals with the fact that breakers may be tripped and power flow interrupted for natural events and for maintenance, as well as by malicious interference, and notes that a good Intrusion Detection System (IDS) should be able to tell the difference. It is with this in mind that the energy datasets in question involve maintenance events and "natural" faults as well as normal operation and attack scenarios. A concise overview of machine learning classifier algorithms and their outcomes for the dataset is also provided in [4]. However, timestamp data was explicitly removed from the energy dataset for the purposes of their experiment, and the focus of the paper is solely on the power states within the experimental setup that was modelled, incorporating no other contextual data. The current work will seek to extend that evaluation by incorporating contextual data which should allow better evaluation of the plausibility of an actual physical fault where anomalous data may otherwise be classified as suspicious – that is, to improve the accuracy of the identification of malicious interference by further helping to eliminate false positives. A paper by Anwar et al. [2] makes an explicit claim that work on IDS for power systems until that time had ignored the presence of actual physical faults. This was not strictly true, but they do accurately point out that the majority of work in the field relied simply on detecting anomalies or unexpected deviations away from baseline values of normal behaviour, and fail to adequately distinguish physical faults from attack conditions. The strength of their work is that they consider more complex fault types than in [4], going beyond tripped breakers to look at measurements resulting from different types of shorting faults. They use the same datasets as their starting point, and work to improve accuracy of classification through the use of Principal Component Analysis (PCA). Wei and Mendis in [21] evaluate a relatively new machine learning technique, Deep Belief Network, and its extension Conditional DBN, against the more usual Artificial Neural Network techniques for evaluation of complex patterns of power system features. A strength of this work is that it not only looks for a "physical coherence" of measurements between components of the system, it is also capable of tracking the change over time to aid in the verification of the current state and the identification of corrupted values being fed into the system. This, however, may in fact make the algorithm more prone to the false positive identification of sudden physical faults as being malicious attack. Kosek [11] uses the more conventional Artificial Neural Network (ANN) algorithms for anomaly detection, but the paper usefully looks at the differences between "point" and "contextual" anomalies – the former involving simply identifying measurements which are anomalous in the global setting, and the latter being able to distinguish measurements which would be anomalous in certain contextual settings but acceptable in others. The latter requires state awareness over time and is more accurate, and in that respect confirms some of the above-referenced work; usefully for the current project, it looks at the addition of meteorological data and timestamping as inputs. The paper, however, concentrates primarily on the customer consumption side and local PV generation, rather than looking at substation models, but the techniques for using contextual data can potentially be applied. The authors of Tian et al. [18] raise valid concerns about anomaly detection and the security of sensor data and PMUs in the real world. They legitimately point out that this is complicated by the fact that an attacker may also be monitoring configuration changes and reacting dynamically; however, it does point to interesting work to be done in the future. Their work is weak in that their experimental model is in a number of respects dissimilar from the physics of the real world, and it goes outside the scope of the present project, but it does raise ideas that future researchers would be advised to take note of for real-world mitigation of data injection attacks against sensor measurements.

III. Methodology

This particular research focuses on a potential situation where a compromise has occurred which results in spoofed or altered data being fed to sensors to induce inappropriate grid behaviour. We look at the values reported as behaviour data from components of a small simulated grid (two power generators, four Intelligent Electronic Devices (IEDs), and four breakers, arrayed along two lines, as illustrated in Fig. 2).

On Fig. 2, G1-G2represent generators, BR1-BR4 - breakers, IED1-IED4 - Intelligent Electronic Devices (IEDs) that provide control of breakers, and B1-B3 indicate power buses where Line 1 spans from B1 to B2, and Line 2 spans from B2 to B3. Here, each IED controls one breaker.

The components of a Smart Grid may be said to have the following "legitimate" operational states:

- normal operation;
- normal (pre-scheduled) maintenance;
- abnormal operation as response to physical component damage or physically anomalous situations;
- recovery from damage or an abnormal state.

Malicious injected data may impersonate either the demands of "normal operation" (for example, indicating that there is increased or decreased demand on certain circuits, inducing an "appropriate" response which in fact results in either overload or underloading), or can be used to indicate component damage where components are in fact operating normally, again thus inducing a mismatched grid response. This paper looks specifically at using contextual analysis involving multiple sensor data, weather event data and developed profiles of normal behaviour, in a machine learning tool for the detection of anomalous component behaviour in general and the evaluation of anomalies in an extended context to best distinguish between legitimate grid response to a likely physical event (such as a lightning strike which damages or disables a grid component) and illegitimate manipulation. Behaviour

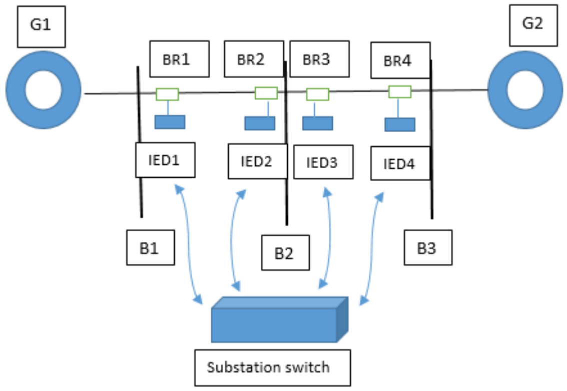

Fig. 2. Modern Substation Components

represented in the datasets includes both “natural” short-circuit faults and malicious data injection attacks targeting voltage or current control.Datasets of electricity substation sensor data under different conditions (including those of malicious data injection), in combination with a modified ”lightning strike event” dataset tested for temporal and geographic correlation (representing proof of concept that weather event data can be usefully incorporated into behaviour analysis) are used to develop and train a bespoke machine learning algorithm which is capable of detecting malicious data injection on smart grids, and distinguish that from weather related events.

A. Datasets of electricity substation sensor data

The datasets around which this work was based were gained primarily from the repository made public by Tommy Morris, in cooperation with others at the Mississippi State University and Oak Ridge National Laboratory in the USA [4]. These sensor and network datasets encompass the miniature model substation components detailed above and were provided as 15 initial sets encompassing a mix of 37 different “natural” and “attack” events in each. They are provided in .csv or ARFF format, and were modified for our purposes by adding generated timestamps and removing fields that did not represent sensor measurement data. The datasets collected from [4] included multiple scenarios coded as shown in Table I.

TABLE I
ENERGY DATASET EVENT SCENARIOS

Scenarios	**Descriptions**
	Natural Event (Short-Circuit Faults)
1	short-circuit on Line 1
2	short-circuit on Line 1
3	short-circuit on Line 1
4	short-circuit on Line 2
5	short-circuit on Line 1
6	short-circuit on Line 1
	Data Injection Attacks – Short-Circuit Fault Replay
7	short-circuit on Line 1 to force tripping command
8	short-circuit on Line 1 to force tripping command
9	short-circuit on Line 1 to force tripping command
10	short-circuit on Line 2 to force tripping command
11	short-circuit on Line 2 to force tripping command
12	short-circuit on Line 2 to force tripping command
	Maintenance
13	Line 1 Maintenance Down
14	Line 2 Maintenance Down
	Normal Operations (No Events)
41	Normal Operational Load Changes

The dataset includes 29 types of measurements (see Table II) for each of four phasor measurement units (PMUs), where each PMU is associated with one of the IEDs. This results in a dataset with a total of 116 features.

TABLE II
ENERGY VALUE MEASUREMENTS

Feature	**Descriptions**
PA1: VH – PA3: VH	Phase A – C Voltage Phase Angle
PM1: V – PM3: V	Phase A – C Voltage Phase Magnitude
PA4: IH – PA6: IH	Phase A – C Current Phase Angle
PM4: I – PM6: I	Phase A – C Current Phase Magnitude
PA7: VH – PA9: VH	Pos.-Neg.– Zero Voltage Phase Angle
PM7: V – PM9: V	Pos.-Neg.– Zero Voltage Phase Magnitude
PA10: VH – PA12: VH	Pos.-Neg.– Zero Current Phase Angle
PM10: V– PM12: V	Pos.-Neg.– Zero Voltage Phase Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA: Z	Appearance Impedance for relays
PA:	ZH Appearance Impedance Angle for relays
S	Status Flag for relays

The original records in these datasets were apparently timestamped as series data, however by the time the datasets were posted to the public repository all timestamps had been removed. This being the case, a column was added and timestamps were artificially generated, spaced out at every 30 seconds. This is not a realistic scenario, as measurement sampling would realistically take place either on a much faster and narrower timescale, or on a wider one via a measurement aggregator. For proof of concept this was deemed sufficient, however. A column was also added to encompass a class value of 0-3 for the overall class of the scenario, which would be used for data analysis in the machine learning algorithms (Table III).

TABLE III
CLASSIFICATION SETS

Class	**Type**	**Scenarios**
0	Normal Operation	41
1	Maintenance	13, 14
2	‘Natural’ Fault	1, 2, 3, 4, 5, 6
3	Attack	7, 8, 9, 10, 11, 12

Finally, a column was added to the features of the dataset to allow indication of correlation with a lightning strike (the dataset described next), coded with a value of 0 for no correlation, 1 for a correlation.

1) External weather datasets: In a live situation, lightning data could potentially be harvested from NOAA-associated organisations, lightning-monitoring bodies which sell monitoring subscriptions such as Vaisala [19], or UK lightning strike data which is publicly available. For the purposes of this work, however, it was decided to simulate lightning strike data in the form of generating records of ”strikes” with random locations and timestamps, within certain constraints. Since the training and test data set for grid events is from a simulated network without a real location, an arbitrary latitude and longitude location was assigned to it for the purposes of the experiment, and this assigned location was then used as the central coordinate for generated random ”lightning strikes.” Values for generated geographical locations of ”strikes” were constrained to within ±0.000009 degrees latitude and ±0.0000005 degrees longitude of the base location. In the real world, collection of lightning strike data would likely be somewhat constrained to relevant areas, but would still contain lightning strikes not near enough to grid components to be relevant, so that some form of pre-processing would be required to screen out irrelevant lightning events. For the purposes of testing our proof of concept we generated a relatively small number of lightning strikes overall, but within that small number there still needed to be enough generated values which corresponded closely to the precise location of the substation. In order to ensure this, the considered geographic coordinate spread could not be overly wide. In addition, a deliberate skew was made in the Python code used to generate the coordinate and timestamp data to ensure that lightning strike events should correlate more with the "natural fault" events noted in [4], rather than with other event categories. It would have been simpler to randomly designate a minimum number of "natural" fault event records as being correlated to a lightning strike; however, the preprocessing of an additional lightning event dataset to test for potential geographical and temporal correlation with fault events was deemed valuable by the researchers as something similar would need to be done in any live situation, as noted above. Thus, the decision was made to generate the random "lightning strikes" and process these for correlation with the energy dataset events in a "normal" fashion. Screening the generated "lightning strike" dataset for potential correlations involved a simple test of records for location within ±0.000001 latitude and ±0.000001 longitude of the base location, and occurrence within the 5 seconds previous to the timestamp of any given measurement within the energy dataset.

In some work that has already been done in the field with machine learning algorithms for the detection of anomalous states, time-series measurements were used to build a profile of what normal deltas would be, and Markov Models used to flag deviations [1]. We are proposing ANN-based algorithms for this detection and flagging of anomalous states, described next.

B. Data Pre-processing

To prepare the data prior to develop any model, in this study we used to novel approaches of Outlier Removal and Missing data replacement.

1) Outlier Removal: By removing the noise from the dataset we are technically removing the insignificant features from the dataset. When we have the significant features within the dataset the next step is to normalise the dataset. Outlier in input data not only can skew and mislead the training process but also can increase training time significantly. However one of drawbacks is to have the required information about the parameters and their distribution before trying to reduce the outlier data. Which not usually something viable and usually visual tools are used to visually identify the distribution in the dataset. Whereas when we are developing a framework this issue need to be dealt with automatically. As it was discussed earlier a distance-based approach was introduced to overcome this statistical approach. In particular [3] defines distance-based approach into 9 phases of :

- Data collection
- Compute the distances of each data
- Identifying maximum distance value of data
- Determining threshold distance value using identified maximum distance
- Compare between threshold distance value and distance of each data
- Determine threshold value(t)
- Determine the distance in comparison to threshold
- Test and identify outlier
- Use Manhattan Distance Technique (MDT) to analyse the data

MDT is used for single dimension data which is used to identify the sum of the absolute distance between elements of parameters (see equation 1).

$$d(t_i, t_j) = \sum_{h=1}^{k} \left|(t_{ih} - t_{jh})\right| \tag{1}$$

We will calculate MDT for each parameters using Scikit-learn library.

Then the following steps will be taken to remove outlier data from the dataset:

- select the attributes where MDT is higher than the average distance of the parameter elements.
- set predefined replacement value. the predefined replacement value is "NAN". This value will be used to mark records with outlier values. Although this feature could be used to eliminate outlier data but sometimes available data is so important and limited that records cannot be simply removed by having outlier data.Therefore this parameter for such cases could be set to "MISSING" and handled in the next step of missing data replacement.
- scale down each of the selected parameters between 0 and 1
- calculated the standard deviation σ for each parameter elements(see equation 2)

$$\sigma = \sqrt{\mu_2} \tag{2}$$

- compare the element value to standard deviation. if distance is more than the default value of 0.3 (value can be modified) then mark the parameter with the predefined value.
- scale back the parameters to original
- remove records with the predefined attributes of "NAN".

2) Missing Data Replacement: In machine learning, although missing data can be cause by the faulty sensor or human error, but sometimes it could potentially be the expected value. According to [13] Missing data values can be divided into two types: "(1) values that are missing at random or for reasons unrelated to the task at hand", "and (2) values whose absences provides information about the task at hand". Therefore it is important before trying to replace or remove a missing data we need to first understand if the missing data is representing a lack or information or it is caused by a fault or an error. In this study we do not deal with the case two scenario, where missing data is actually representing an information. The main focus of this study on the first scenario where the missing data caused due to a problem and it requires to be replaced. Some of the most used approaches to deal with missing data is already discussed in the literature review chapter which includes MLP, SOM and KNN. Amongst those KNN has been adapted as preferred method of dealing with the missing data.

To replace the value of the parameters with the assigned value of ”MISSING” from the Outlier removal phase, we feed in the data into a KNN model.The model will find the closest neighbours using distance metric to ultimately replace the MISSING value. The following steps are taken to replace the missing value:

- Select all the parameters which include ”MISSING” values.
- for each parameter we create a list of values excluding ”MISSING” values.
- the filtered array for each parameter is given to KNN model to find the closest neighbours using according to a distance metric.
- from the original dataset we find the neighbouring elements of the ”MISSING” .
- we then find the clusters the neighbouring values belong to and we calculate the weighted average of the neighbouring clusters to replace the ”MISSING”.

It means if for instance x is n_{th} element of an input array (i.e., $m_n = 1$) which is missing , once k which is the nearest neighbours to the element identified then x is estimated using corresponding n_{th} feature value of ν. [8].

$$\nu = \{v_k\}^K{}_{k=1} \tag{3}$$

Moreover, after removing noises and replacing missing data, data were altered furthur as follows:

- Removed the features for SNORT and logs, as we are looking at line and component data only.
- Added and populated a broad category classifier column (0-3)
- Added timestamps to the rows for tests for potential lightning correlation.
- Added and populated a feature for potential lightning correlation.
- ‘inf’ values replaced by a default value of -999999.
- Removed the timestamps.
- Removed the scenario column and separated the classification column from the datasets to be used as the labels for testing and training.

The machine learning algorithms deemed to be most suitable in the chosen problem domain are Support Vector Machines, as these can deal with high-dimensional data. Also in recent years it is proven that deep neural networks can cope well with multi-dimensional datasets and capable of forming a reliable and effective models. Firstly, the substation sensor dataset was analysed using the SVC algorithm supplied by Python’s Scikit Learn module with default hyper-parameter values (‘RBF’ kernel, no class weighting and C=1.0). Secondly, the dataset underwent the analysis through a bespoke developed neural network model. To prepare the data fit for both SVM and ANN machine learning algorithms, data cleaning, conversion, normalisation and reduction have been applied to the chosen dataset. During the data cleaning stage as well as replacing null, text and missing values with numbers, dataset record also needed to be scaled down to the range between 0 and 1.

In addition, the class column from the dataset contains values ranging from 1 to 14 with a further class of 41. To be able to classify the dataset in a steady format, a data encoder is used to transform the class labels between the range 0 to 15. (see Algorithm 1)

Algorithm 1 Data Processing

```
data Processing (dataset)
*UnitToDrop* ← 35%
repeat
    for $i \leftarrow 1, rows$ do
        Outlier removal
        Missing data replacement
        scale down values to the 0 to 1 range
    end for
until data is scaled and normalised
Split Training and Test based on UnitToDrop
repeat
    Reshape Training Dataset
    for $i \leftarrow 1, rows$ do
        encode classification values
        range classification from 0 to 15
    end for
until training and test datasets are reshaped
Return (trainingDataset, testDataset)
```

The datasets were shuffled and partitioned for training using Pandas and Numpy library, with 65% of each dataset assigned for training and 35% of each dataset assigned for testing. Moreover, to train the models further, additional libraries Scikit Learn, Tensorflow and Keras have been used. For the visualisation, i.e. generation of the confusion matrices and visual diagrams, the matplot library has been used.

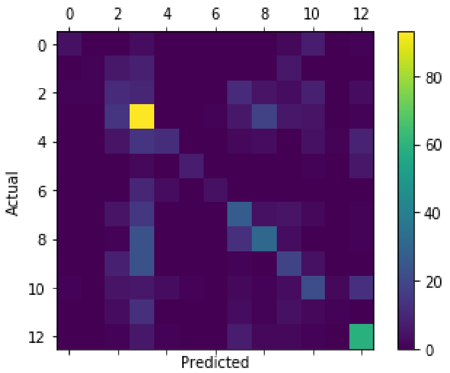

Fig. 3. SVM Confusion Matrix

IV. Results

In this section we compare two supervised machine learning algorithms applied to data analysis of malicious signal manipulation in Smart Grid systems.

A. Support Vector Machine

The first run, Dataset 1 through a default python SVC Support Vector Machine, achieved an accuracy of 41.3%. Running the dataset through again with default algorithm settings, but with all 'inf' records removed, boosted the accuracy to close to 44%, and expanding C to 3.0 rather than 1.0 to make classification borders somewhat harder, boosted accuracy to 45%, with the following confusion matrix (see Fig. 3 3).

The skew towards a diagnosis of "natural fault" is obvious. What is not immediately obvious is why - which can potentially become an issue requesting further investigation. When the algorithms were run with all the rows containing 'inf' values removed (although that accounted for several hundred records out of each of the 15 original datasets) rather than given substitute values, in general that added an additional 2-4% accuracy to the accuracy scores achieved, so this may be counted as marginal but not spectacularly successful. Since missing or out-of-bounds values are unfortunately a common occurrence in real-world sensor sampling, however, this is undoubtedly an issue that could be better addressed, although no suggestions presented themselves at the time of writing. The larger, consolidated dataset C2 run through the above SVM achieved 43.4% accuracy, probably largely on the strength of the greater number of records available in the dataset for training.

B. Artificial Neural Network (ANN)

To achieve better results in this study by improving the accuracy of the SVM model, a new bespoke neural network model has been developed. The model uses total of 6 dense/fully connected layers in conjunction with dropout technique before the last dense layer.

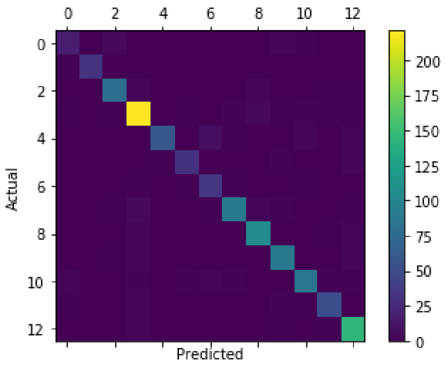

Fig. 4. ANN Confusion Matrix

Prior to using dropout technique, the model was generating elevating loss values, which clearly indicated overfitting. However, by deploying dropout, loss value gradually and steadily decreases. Dropout is a technique proposed by Srivastava [17], where a random proportion of the neurons in a layer are dropped during training. The fact that they are "dropped-out" randomly means that their contribution to the activation of downstream neurons is temporally removed to avoid overfitting the model. Through multiple trials the model developed for this study, can generate an accuracy of 98.8%, over a total of 1000 iteration.

Figure 4 shows the confusion matrix of the tested dataset against the developed model.

The model can generate high accuracy eventually, but to get to such accuracy a large number of training iterations (the x-axis on the figures) are required. The model accuracy can potentially improve with even more training iterations. However, for this study the model was trained over 1000 iterations. Figure 5 illustrates the gradual improvement of the model accuracy.

Also Figure 6 shows the gradual error elimination and gradual decrease of loss rate, which indicate that the model is not suffering from overfitting, and continued training over time can gradually improve the model's accuracy and reduce the loss value.

V. Conclusion and Discussions

The growing complexity of Smart Grids necessitates exploration of various approaches to monitoring, optimisation, and more importantly, securing modern power generation and distribution systems. These new approaches need to provide advanced data analytical capabilities for making sense of the data, situational awareness, and the capacity to quickly detect, and ultimately withstand, abnormal conditions - both natural and of malicious nature.

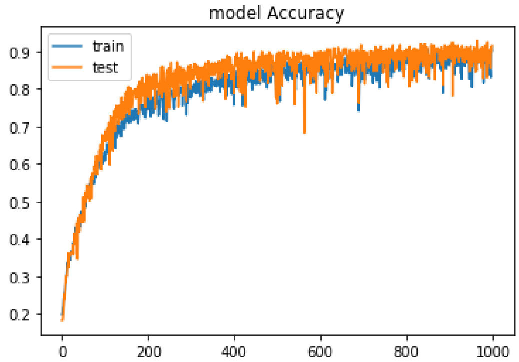

Fig. 5. ANN Model Accuracy

In the present study we suggested an approach to effective detection of operational anomalies in Smart Grid systems based on machine learning algorithms that are capable of processing both substation sensor and contextual data. Although the performance of the chosen algorithms substantially differ, the ANN-based approach gives us a powerful and quite effective tool for achieving the desired outcome, but is best done with a technique which controls overfitting, provided here by dropout. Testing needs to be extended to further situations to determine the degree of value added by the inclusion of lightning or other contextual weather data. In addition, in realistic situations an energy measurement dataset would be heavily skewed towards normal operational states, but even the number of faults would be many orders of magnitude larger than the number of lightning strikes in an area, so work must be done to manage how the algorithms may be trained in the presence of extraordinarily unequal datasets.

Overall, there are a number of steps that could be taken in future extensions of this particular research project. First and foremost among them would be an exploration of the type of Principal Component Analysis done in [6], in order to identify more exactly what the highest-value features of the datasets are and to allow dimension reduction. In addition to that also the developed neural network model can be retrained with additional dataset and can go live on production using platform such as Tensorflow Serving to detect malicious signal manipulations on real-time.

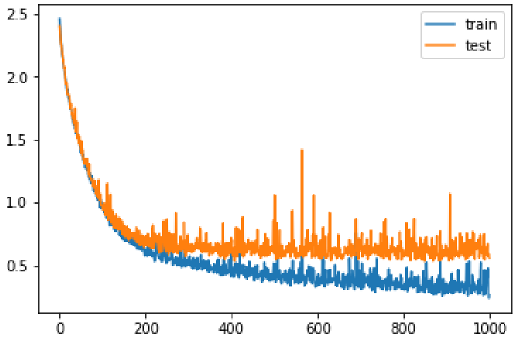

Fig. 6. ANN Model Loss

Acknowledgement

Special thanks for the energy datasets need to go to the Oak Ridge National Laboratory, and especially Raymond Borges Hink, who was especially helpful in correspondence.

References

[1] Anderssen, M. Modeling electricity load curves with hidden markov models for demand-side management status estimation. *International Transactions on Electrical Energy Systems 23* (09 2016).

[2] Anwar, A., Mahmood, A. N., and Shah, Z. A data-driven approach to distinguish cyber-attacks from physical faults in a smart grid. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (2015), ACM, pp. 1811–1814.

[3] Bakar, Z. A., Mohemad, R., Ahmad, A., and Deris, M. M. A comparative study for outlier detection techniques in data mining. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on* (2006), IEEE, pp. 1–6.

[4] Beaver, J. M., Borges-Hink, R. C., and Buckner, M. A. An evaluation of machine learning methods to detect malicious scada communications. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on* (2013), vol. 2, IEEE, pp. 54–59.

[5] Heussen, K., Tyge, E., and Kosek, A. M. Residential demand response behaviour modeling applied to cyber-physical intrusion detection. In *PowerTech, 2017 IEEE Manchester* (2017), IEEE, pp. 1–6.

[6] Hink, R. C. B., Beaver, J. M., Buckner, M. A., Morris, T., Adhikari, U., and Pan, S. Machine learning for power system disturbance and cyber-attack discrimination. In *2014 7th International symposium on resilient control systems (ISRCS)* (2014), pp. 1–8.

[7] Isozaki, Y., Yoshizawa, S., Fujimoto, Y., Ishii, H., Ono, I., Onoda, T., and Hayashi, Y. Detection of cyber attacks against voltage control in distribution power grids with pvs. *IEEE Transactions on Smart Grid 7*, 4 (2016), 1824–1835.

[8] Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., and Franco, L. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine 50*, 2 (2010), 105–115.

[9] Ju, P., and Lin, X. Adversarial attacks to distributed voltage control in power distribution networks with ders. In *Proceedings of the Ninth International Conference on Future Energy Systems* (2018), ACM, pp. 291–302.

[10] Kosek, A. M. Contextual anomaly detection for cyber-physical security in smart grids based on an artificial neural network model. In *2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG)* (2016), IEEE, pp. 1–6.

[11] Kosek, A. M., Gehrke, O., and Kullmann, D. Fault tolerant aggregation for power system services. In *Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on* (2013), IEEE, pp. 107–112.

[12] Laboratory, O. R. N. Dragos security team, “crashoverride: Analyzing the threat to electric grid operations, June 2017. https://dragos.com/blog/crashoverride/CrashOverride-01.pdf.

[13] Libbrecht, M. W., and Noble, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics 16*, 6 (2015), 321.

[14] Mo, Y., Kim, T. H.-J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., and Sinopoli, B. Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE 100*, 1 (2012), 195–209.

[15] Morrow, K. L., Heine, E., Rogers, K. M., Bobba, R. B., and Overbye, T. J. Topology perturbation for detecting malicious data injection. In *2012 45th Hawaii International Conference on System Sciences* (2012), IEEE, pp. 2104–2113.

[16] Petrovski, S., Malakhov, A., Kopyriulin, P., and Petrovski, A. Adaptation of smart grid technologies: The use of computational intelligence for reliability estimation and maintenance scheduling./in the proceedings of the world congress on computational intelligence, wcci 2012, brisbane, australia. fuzzieee, cat.: Cpf12fuz-usb. *FUZZIEEE, Cat.: CPF12FUZ-USB, ISBN* (2012), 978–1.

[17] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*, 1 (2014), 1929–1958.

[18] Tian, J., Tan, R., Guan, X., and Liu, T. Hidden moving target defense in smart grids. In *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids* (2017), ACM, pp. 21–26.

[19] VAISALA. Global lightning dataset gld360, Nov. 2018. https://www.vaisala.com/en/products/data-subscriptions-and-reports/data-sets/gld360.

[20] Wang, J., and Peng, C. Analysis of time delay attacks against power grid stability. In *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids* (2017), ACM, pp. 67–72.

[21] Wei, J., and Mendis, G. J. A deep learning-based cyber-physical strategy to mitigate false data injection attack in smart grids. In *Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Joint Workshop on* (2016), IEEE, pp. 1–6.