Received September 10, 2020, accepted October 10, 2020, date of publication October 19, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032251

Behavioural Intrusion Detection in Water Distribution Systems Using Neural Networks

TSOTSOPE DANIEL RAMOTSOELA[1], (Member, IEEE), GERHARD PETRUS HANCKE[2], (Senior Member, IEEE), AND ADNAN M. ABU-MAHFOUZ[1,3], (Senior Member, IEEE)

[1]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa
[2]Department of Computer Science, City University of Hong Kong, Hong Kong
[3]Council for Scientific and Industrial Research (CSIR), Pretoria 0184, South Africa

Corresponding author: Gerhard Petrus Hancke (ghancke@ieee.org)

This work was supported in part by the Research Grants Council of Hong Kong through Project CityU 21204716 and in part by City University of Hong Kong through Project CityU 11202618 (7005053).

ABSTRACT There has been an increasing number of attacks against critical water system infrastructure in recent years. This is largely due to the fact that these systems are heavily dependent on computer networks meaning that an attacker can use conventional techniques to penetrate this network which would give them access to the supervisory control and data acquisition (SCADA) system. The devastating impact of a successful attack in these critical infrastructure applications could be long-lasting with major social and financial implications. Intrusion detection systems are deployed as a secondary defence mechanism in case an attacker is able to bypass the systems preventative security mechanisms. In this thesis, behavioural intrusion detection is addressed in the context of detecting cyber-attacks in water distribution systems. A comparative analysis of various predictive neural network architectures is conducted and from this a novel voting-based ensemble technique is presented. Finally an analysis of how this approach to behavioural intrusion detection can be enhanced by both univariate and multivariate outlier detection techniques It was found that multiple algorithms working together are able to counteract their limitation to produce a more robust algorithm with improved results.

INDEX TERMS Anomaly detection, cyber-physical security, industrial control system, machine learning, water distribution system.

I. INTRODUCTION

The deployment of cyber-physical in critical infrastructure applications such as water distribution systems is going to be paramount in realising the envisioned smart cities of the future [1]–[3]. These cyber-physical systems conveniently allow for the remote monitoring and control of the physical process by allowing the industrial control network to incorporate both internal and external communication. This means that industrial control networks are now similar conceptually to conventional computer networks but remain structurally different owing to the conflicting priorities of the two systems [4]. This however also means that these networks are now as vulnerable to the same security threats that plague conventional IT networks with far more severe implications. The increased number of cyber-attacks against these critical infrastructure applications in recent years is evidence of this and the consequences of them being successful could be devastating for the people that rely on them and in some cases also the natural environments where they are deployed [5].

An intrusion occurs when an attacker bypasses these external security mechanisms in an attempt to compromise at least one of the three key pillars of network security (confidentiality, integrity and availability) [6]. When these preventative security mechanisms fail, detection mechanisms, diversionary tactics and countermeasures can be used to limit the potential damage an attacker can cause [7]. This group of security mechanisms and protocols are collectively referred to as Intrusion Detection and Prevention Systems. An intrusion detection system (IDS) is a security mechanism consisting of software and/or hardware that

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed M. Elmisery.

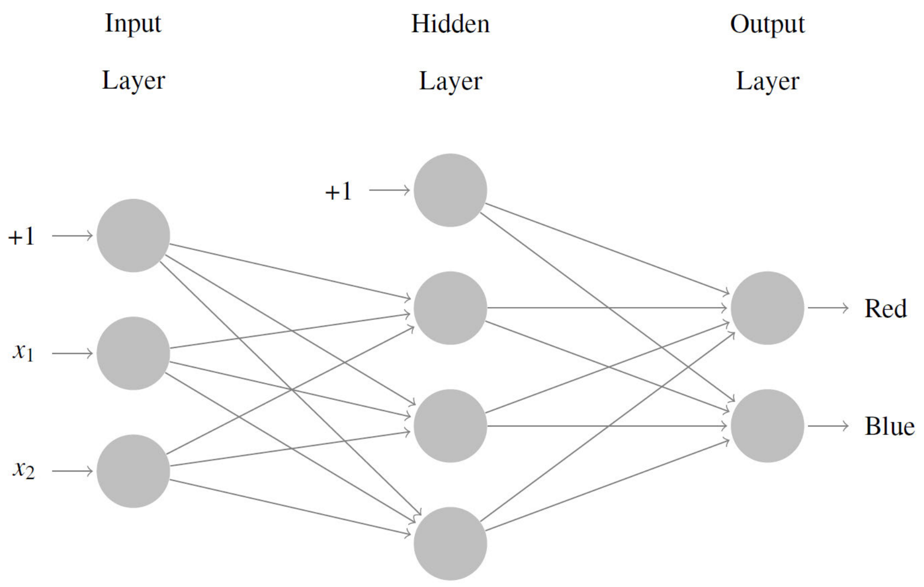


FIGURE 1. Typical structure of ANNs.

attempts to autonomously detect malicious behaviour within the system [8].

In this work we focus on behavioural IDSs which are preferred over traditional signature-based methods because they provide greater detection generality [9]. This generality however comes at a cost of accuracy because these systems generally have a lower precision than their signature-based counterparts because they normally have significantly more false positives. This is acceptable when protecting critical infrastructure because the reactive nature of signature-based methods is not ideal because the consequences of successful cyber-attacks could be devastating [10]. Behavioural IDSs are specialised versions of anomaly/outlier detection systems was pioneered by the statistics community in its early days of inception [11]. A major challenge in these outlier detection applications is that there is an imbalance of data as outliers by nature will make up a small proportion of any dataset. This means that there will be a shortage of positive samples to adequately train machine learning models.

Traditional outlier detection techniques aren't concerned with the system dynamics but rather in trying to directly distinguish between normal and anomalous data instances. In this paper, a second approach to outlier detection is considered where the model attempt to learn normal system behaviour and predicting particular elements. If the error between the predicted and actual values is beyond some predetermined threshold then a data instance is considered anomalous. The main contributions of the work presented in this paper are as follows:

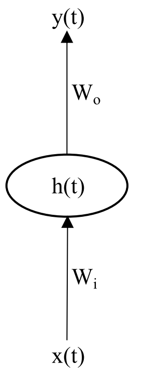
1) A number of popular neural networks are used to learn the normal behaviour of the system and predict tank levels within the BATADAL dataset. The Mahalanobis distance is then used as the second stage statistical technique distinguish between normal and anomalous data.
2) A voting-based ensemble technique is proposed to create a more robust algorithm that improves the overall performance by leveraging the strengths of each individual algorithm. In essence the goal of the algorithm is to improve the performance metrics that evaluate the detection accuracy.

The rest of this paper is organised as follows: First the different neural network architectures that were implemented are discussed (Section II). The chosen anomaly detection approach is then discussed followed by the proposal of a novel voting-based ensemble technique for anomaly detection using the implemented algorithms (Section III). The results of our experiments are then presented and discussed (Section V). Finally, some observations from the results are discussed (Section VI) and the paper is concluded (Section VII).

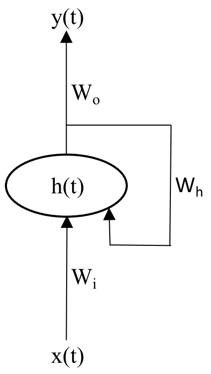
II. NEURAL NETWORKS

A. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) have the ability to model very complex input/output relationships which have made them very popular in practical applications which range from self driving cars to fraud detection in the financial sector [12]. ANNs present a black box approach to machine learning where the algorithm captures very complex relations without input from the user about the structure of the output. The basic structure of an ANN consists of a number of interconnected artificial neurons or nodes that can be categorised into three main layers as shown in Figure 1. The network has one input layer which has a node count that is equivalent to the number of inputs. The network can also have multiple hidden layers with the user rationally selecting the number of layers and nodes in each layer depending on the complexity of the problem. Finally the network has an output layer which has the same number of nodes as the number of outputs.



(a) ANN Configuration



(b) RNN Configuration

FIGURE 2. Difference between ANN and RNN configurations.

In classification problems the node count in the output layer could correspond to the number of categories in the dataset. To get the output of a particular node j, the weighted sum of inputs from the previous layer and a bias value are put through an activation function as shown in (1), where g is the activation function, b is the bias value and w is the weight association between two nodes [13].

$$a_j = g(b_j + \sum_{i=1}^{\ell} a_i \cdot w_{i,j}) \tag{1}$$

B. RECURRENT NEURAL NETWORKS

The sequence of events in water distribution systems is important in evaluating whether or not an anomaly has occurred. For instance, consider a tank in one time step that is at 80% capacity but in the following time step the same tank is at 40% capacity and that the interval between time steps is 10 minutes. The two tank levels and input parameters could be within the normal operating range of the system but the large reduction level reduction within one time instance is an anomalous event within the context of the system. These types of anomalies are called contextual anomalies. Recurrent neural networks (RNNs) are neural network architectures that are able to capture the temporal dynamics of the system unlike conventional ANNs where each time instance is viewed as independent of all other instances.

RNNs are able to predict future states not only based the current input instance, but also taking into account previous systems states [14]. This essentially means that the RNN has a memory of what was happening previously in the system and includes this contextual information when making predictions. The difference between the structure of the feed-forward ANNs and RNNs is shown in Figure 2. For the ANN the input nodes is connected to a hidden node and its contribution is controlled with weight W_i. The hidden node also has a weighted connection W_o to the output node. The RNN structure is very similar except that the hidden node has a feedback loop with a weighted contribution of W_h. What this means is that like the ANN the weighted input is passed through the hidden node as an additional weighted input. The output of the hidden layers for both ANN and RNN are shown in (2) and (3) respectively. The only difference between the two equations is the additional weighted input from the previous state.

$$h(t) = g(b_i + W_i x(t)) \tag{2}$$
$$h(t) = g(b_h + W_i x(t) + W_h h(t-1)) \tag{3}$$

The RNN architecture shown in Figure 2 is the most basic RNN configuration and is known as the simple or Elman RNN [14]. The Elman RNN is a powerful algorithm that has been shown to produce great results in a variety of different application. The main issue with this configuration is that it is structured such that earlier time steps get diluted over time. This means that the configuration prioritises short-term memory and "forgets" earlier instances over time. In some applications this property is not an issue and could even be preferred but in others it could seriously affect the performance. It is for this reason that more complex RNN architectures that possess the ability to retain long-term memory have been proposed. In this work we consider only two of the most popular of these architectures which are described in the sections that follow.

1) LONG SHORT-TERM MEMORY

The major issues with the basic RNN architecture are the vanishing/exploding gradient problem and also its inability to retain long-term memory. Long short-term memory (LSTM) networks were designed specifically to deal with these issues and have been one of the most effective RNN architectures proposed to solve these problems [15]. LSTMs are structured similarly to the basic RNN except that the hidden node characterised by (3) is replaced by a more complex memory cell. This memory cell provides it with the ability to retain long-term memory and solves the vanishing gradient problem by not directly using a non-linear function to alter the state of the cell. This means that the back-propagation algorithm does not need to be artificially constrained so the error can be distributed without restrictions.

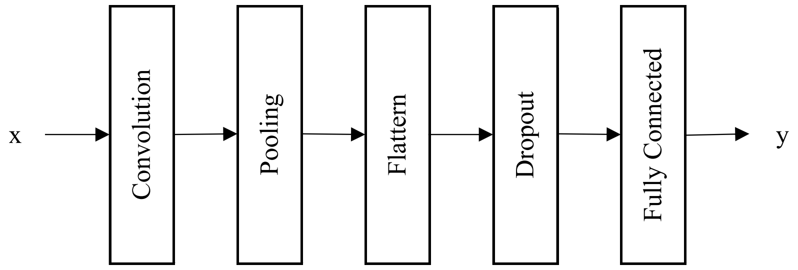


FIGURE 3. Layers of a Typical CNN Architecture.

The LSTM memory cell consists of three gates which control the proportional contribution of the different parameters [16]. Each of the gates takes in as input the previous output of the cell and the current input and their operations are analogous to the reset, read and write operations in physical memory chips. The forget gate $f(t)$ controls how much the previous state will affect the new cell state. The input gate $i(t)$) on the other controls how much this state will be affected by the input to the cell. The input and forget gates thus collaborate to produce the new cell state $c(t)$. The output gate $o(t)$ controls how the cell state will affect the output of the cell $h(t)$. The relationships just described are shown in (4) meaning that unlike the basic RNN which is controlled by one equation, the output of LSTM is dependent on five equations.

$$\begin{aligned} f_t &= \sigma(b_f + W_{xf}\, x_t + W_{hf}\, h_{t-1}), \\ i_t &= \sigma(b_i + W_{xi}\, x_t + W_{hi}\, h_{t-1}), \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh(b_c + W_{xc}\, x_t + W_{hc}\, h_{t-1}), \\ o_t &= \sigma(b_o + W_{xo}\, x_t + W_{ho}\, h_{t-1}), \\ h_t &= o_t \cdot \tanh(c_t) \end{aligned} \tag{4}$$

2) GATED RECURRENT UNIT

The gated recurrent unit (GRU) is a recent but popular RNN architecture that was proposed by Cho *et al.* [17] in 2014. Conceptually it is very similar to the LSTM but it has fewer parameters and is thus simpler to train. This means that for the same network configuration, GRUs will take less time to train the LSTMs. This simplicity does not however come at a cost of reduced performance as the architecture has been shown to have better or comparable results to LSTMs across a variety of different applications [18]. This means that for most applications LSTMs and GRUs outperform the simple RNN but when compared to each other their performances will vary depending on the applications.

The main difference between the LSTM and GRU is that the latter uses two gates instead of three to control the cell output [14]. The reset gate $r(t)$ determines how much of the previous state is going to affect the new state. The update gate $u(t)$ on the other hand determines how the new input is going to affect the current cell state. Another major distinction between the two architectures is that the GRU does not have an internal cell state. This means that the output is directly dependent on the previous state and new input. How the output of the cell is determined using the described configuration is shown in (5). As with LSTMs, GRUs are not affected by the vanishing/exploding gradient problem with the added benefit that they are simpler to compute.

$$\begin{aligned} r_t &= \sigma(b_r + W_{xr}\, x_t + W_{hr}\, h_{t-1}), \\ u_t &= \sigma(b_u + W_{xu}\, x_t + W_{hu}\, h_{t-1}), \\ \hat{h}_t &= \tanh(b_h + W_{xh}\, x_t + W_{hh}(r_t \cdot h_{t-1})), \\ h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t \end{aligned} \tag{5}$$

C. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN) are biologically inspired but the human visual cortex [19]. Individual neurons react to only specific components of what is within range of the eye. This small parts then overlap with each other to produce one coherent image that covers the entire field of vision. The image is then past on to higher layer neurons which successively produces an increasingly more detailed image. This is the general idea of how a CNN works, the input is broken up into blocks of data which are processed independently and then combined to produce more meaningful data. As the data passes through more and more layers the more useful features start becoming apparent.

CNNs are typically used in image processing applications where images fed into the network as 2D inputs (width × height). This is contrary to traditional ANNs where the input image has to be transformed into a 1D vector before being fed into the network. One of the main reasons CNNs are popular in image processing applications because they are translation invariant [20]. This means that when trying to detect a specific object in an image its location within the image is of no consequence to the algorithm. This property is also useful in many other applications such as speech recognition where translation variance could occur as a result of different speaking styles [21]. This is a convenient property which reduces the pre-processing burden on the users of the algorithm. Even though CNNs are popularly used in image processing applications they are also able to accurately extract the temporal dependencies found in time-series data [22].

Figure 3 shows the main components of a typical CNN architecture. The first layer is the convolutional layer which performs the convolution operation between the input and a filter or kernel. Equation (6) can be used to get the output of this layer at indices $a, b \in o$. The size of the kernel is a hyper-parameter selected by the user usually by conducting experimentation to find the size the produces the best results. The next layer is the pooling layer which aggregates the results of the convolutional layer by performing a downsampling operation on the data. The following layers involve flattening the data into a 1D vector feeding it into a vanilla ANN configuration. A dropout layer can also be added to reduce the size of the network and prevent the overfitting that these kinds of networks are prone to [23]. These main layers can be repeated multiple times in different orders depending on the CNN architecture. It is also possible to replace the fully connected layer with other deep learning layers just as RNN which could greatly improve results depending on the application [24].

$$h(a, b) = ReLu(b_h + \sum_{i \in k} \sum_{j \in k} w(i, j)x(a+i-1, b+j-1)) \quad (6)$$

III. ANOMALY DETECTION

Traditional outlier detection schemes attempt to directly evaluate whether each data instance is an outlier or the degree to which it is an outlier relative to other instances in the dataset. This is similar to how neural networks can be used to predict categorical values instead of an output sequence. In this case the model can be trained on a data set of known attacks and a second dataset that includes previously unseen attacks. The latter would be used to evaluate how well the system generalises. The trouble with using this approach with neural networks is the unbalanced data problem because anomalies are rare obscure events in practice. This means that datasets may generally not have enough anomalous samples to adequately train the model. The result of this could a model with a high precision but low recall rate. Within the context of WDSs where the results of not detecting an attack could be catastrophic, a system with these characteristics would not be desirable. Preprocessing the data could alleviate the problem but in many real-world applications labelled anomalous data is not always available. With ICSs however there is always an abundance of data when the system is working under normal operating conditions so a more attractive solution is to work with prediction as opposed to classification.

The second approach to anomaly detection attempts to model the system dynamics using dependencies to predict known system variables. In this way the model need only be trained on the normal data and the anomalous instances can be used for cross-validation and testing. To detect whether or not an instance is anomalous the error between the actual and predicted value is calculated and in general a threshold can be determined to isolate the anomalies. For this work, the squared error was used because the neural network weights were optimized using the mean squared error (MSE) during training. The cross-validation dataset is then used to find an appropriate threshold and the test dataset can determine how well the chosen threshold generalises.

TABLE 1. Dependency matrix for BATADAL network.

Actuator	Sensor					
	LT1	LT2	LT3	LT4	LT5	LT7
PU1	x	-	-	-	-	-
PU2	x	-	-	-	-	-
V2	-	x	-	-	-	-
PU4	-	-	x	-	-	-
PU5	-	-	x	-	-	-
PU6	-	-	-	x	-	-
PU7	-	-	-	x	-	-
PU8	-	-	-	-	x	-
PU10	-	-	-	-	-	x
PU11	-	-	-	-	-	x

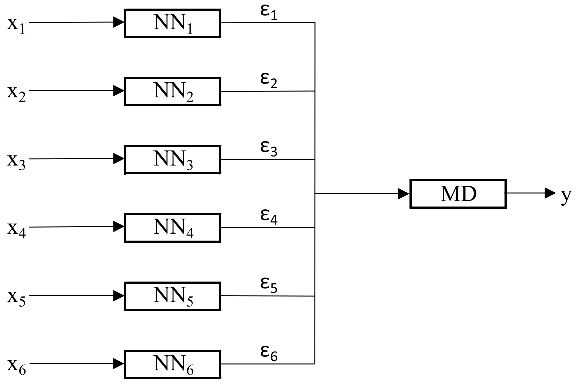


FIGURE 4. NN based anomaly detection scheme.

Instead of using the entire feature space to predict one variable, the system was broken down into multiple subsystems to predict different tank levels. Each tank level is dependent on pumps which have corresponding flow, pressure and status flag variables that can be used as predictors. For the chosen application environment, the sub-systems were chosen based on the dependency matrix shown in Table 1 where *LT* is the tank level, *PU* is a pump and *V* is a valve. The tank levels are different variables which means that each subsystem will have its own error vector which needs to be analysed to find anomalies. This univariate approach to anomaly detection will be able to detect content anomalies on the error vectors but may struggle with contextual anomalies. To account for this a multivariate anomaly detection scheme was used on the entire error matrix which contains error vectors from the different subsystems.

The structure of the proposed system is shown in Figure 4. The NN block represents the different network algorithms that were discussed in the previous section. Each subsystem has its own inputs and produces its own prediction error. All the errors associated with a particular data instance are then fed into a multivariate anomaly detection scheme. The chosen scheme for this system is based on the Mahalanobis Distance (MD) which is preferred over the Euclidean distance because it takes the variability and correlation of the variables into consideration [25]. This is because the Euclidean distance works well when the data is not correlated does but cannot be used to find outliers when it is correlated. The

MD works by first scaling the data to the effect of removing any correlation and then calculating the normal Euclidean distance on the normalised data. In order to scale the data the covariance matrix $C \in \mathbb{R}^{d \times d}$ first needs to be calculated where the covariance between any two features can be found using (7). The MD can be calculated using (8) which is the same as the euclidean equation in (9) except for the covariance matrix used to normalise the data.

$$Cov(x, y) = \frac{1}{\ell - 1} \sum_{i=1}^{\ell} (x_i - \bar{x})(y_i - \bar{y}) \tag{7}$$

$$MD = \sqrt{(\boldsymbol{x} - \mu)^T C_x^{-1} (\boldsymbol{x} - \mu)} \tag{8}$$

$$dist(x, \mu) = \sqrt{(\boldsymbol{x} - \mu)^T (\boldsymbol{x} - \mu)} \tag{9}$$

The MD assumes that the underlying data follows the Chi-Square distribution with d degrees of freedom where d is the dimensionality of the data. The Chi-Square distribution approaches the multivariate normal distribution for larger d values which would be the case which would always be the case in the application scenario. This assumed distribution will not always be correct in the practical set up but the algorithm still works reasonably well even when this is the case. This method is not applicable to high dimensional data due to multicollinearity. This is however not an issue in this application because the input to the algorithm will be an error vector associated with the seven tank levels. In multivariate distance-based methods, the threshold for what constitutes an anomaly is generally selected by the user arbitrarily. For MD however, a probability is used to find a critical value from Chi-Square distribution which is applied to the dataset until an optimal value is found.

A. PROPOSED ENSEMBLE TECHNIQUE

The results of the different implemented neural networks are also combined to create a more robust algorithm that leverages the strengths of each individual algorithm. This is because the algorithms won't consistently perform well in all scenarios but by combining them the ensemble technique will be more robust to the different scenarios. The algorithms were combined through a system of voting as shown in Figure 5. The ensemble technique thus only flags an anomaly if at least two out of the five algorithms agree. This was preferred over the traditional majority rule vote because the application requires a bias to detecting attacks.

The reason for choosing the proposed ensemble technique is that an analysis of the results of the implemented Neural Network techniques showed some inconsistent results across the two datasets. Furthermore, the different algorithms were effective at detecting different attacks but there was a lot of overlap with regards to the data instances that were correctly identified. The algorithms were also unlikely to flag the same data instances as anomalous meaning that errors could be eliminated which would result in significant performance improvement. By using a voting-based approach it is hypothesised that the ensemble technique will be robust and produce consistent results across both datasets while maintaining a good PPV.

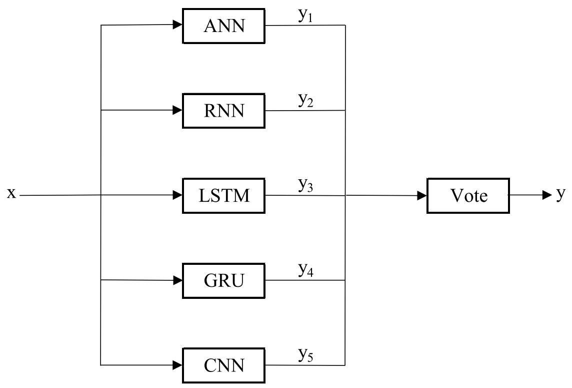


FIGURE 5. Voting based ensemble technique.

As discussed in the previous paragraph, this method is effective in reducing the number of false positives because it is unlikely that multiple algorithms with consistently flag the same normal data instances as anomalous. The technique will however not always produce results that outperform the individual components in all performance metrics but it will be more robust to changing environments and in general have far fewer false alarms. The application scenario is such that the algorithms should be biased towards detection because the cost of not detecting an error is too high. With that being said a system that constantly produces false alarms will be unusable in practice because it will not be taken seriously.

An issue with this approach is that it will result in a computational time that is at least five times that of running an individual algorithm. In time critical applications, this kind of delay would render the proposed technique unsuitable for practical implementation. This is however not the case in this application scenario because of (1) the architecture of the IDS and (2) the machine learning algorithms used. The described IDS uses a centralised architecture meaning that intrusions are detected by a central manager and not the individual nodes. This means it is possible to do all the processing on a central manager with significant computational power instead of using the resource-constrained nodes. The second consideration is that neural networks consume the bulk of their computational load while training and require very little during execution. In fact, even in resource-constrained devices it has been shown that the execution time for deep learning models can be in the $10ms$ range for large configurations [26]. It is for these two reasons that the additional computation burden caused by the ensemble technique will not have a significant impact in this application.

IV. EXPERIMENTAL SETUP

A. BATTLE OF THE ATTACK DETECTION ALGORITHMS

The discussed algorithms were trained and evaluated using the battle of the attack detection algorithms (BATADAL) datasets. BATADAL was a competition where a number of proposed attack detection algorithms were pitted against each

TABLE 2. BATADAL dataset.

Dataset	Normal	Cross-validation	Test
Normal Instances	8761	3676	1682
Attack Instances	0	501	407
Total Instances	8761	4117	2089
Number of Attacks	0	7	7

other on a simulated water distribution system [27]. The competition provided participants with three datasets, one containing only normal data and the other two containing a variety of attacks against the system. One of the attack datasets can be used for cross-validation purposes and the one can be used to test how well the proposed algorithm works when used on unseen data. Table 2 shows the distribution of data in the three datasets provided for the competition. As can be seen in the table, the cross-validation and test datasets each have 7 different attacks represented by 501 and 407 data instances respectively. It can further be seen that the proportion of anomalous vs normal instances in each dataset is approximately 13% and 24% respectively. The structure of the data is thus realistic in that it is unbalanced in favour of normal instances as discussed in Section III. In fact, the proportion of normal instances relative to anomalous instances across all three datasets is approximately 93%. The datasets are also realistic in that the attacks also have corresponding concealment strategies used to make detection even more difficult. In a practical scenario, the attacker would use these concealment strategies so as not to be detected by the system's built-in fault detection mechanisms.

B. ACCURACY METRICS

The main performance metrics from the BATADAL competition have a bias towards detecting attacks because in these systems the consequences of not detecting attacks are far greater than of falsely detecting one. The competition used one overall metric to compare the different algorithms (S score) and that metric is calculated using an accuracy metric (S_{CLF} score) and a time metric (S_{TTD} score). The S_{CLF} is a measure of how effective the system is at detecting attacks and it is calculated using the true positive rate (TPR) and the true negative rate (TNR) as shown in (10). The TPR, also called the sensitivity, is a measure that indicates proportion of detected attacks relative to the total number of attacks in the dataset as shown in (11). The TNR, also called the specificity, is a measure that indicates the proportion of normal instances that were correctly identified relative to the total number of normal instances in the dataset as shown in (12). The S_{TTD} indicates how quickly the algorithm is able to detect attacks and is a measure of when the attack was first flagged relative to when it started as shown in (12). The S combines the S_{CLF} and S_{TTD} scores using the weighting parameter γ as shown in (14). For the competition, γ was given a value of 0.5 indicating that both metrics were given equal weighting in the calculation of the overall S score.

$$S_{CLF} = \frac{TPR + TNR}{2} \tag{10}$$

TABLE 3. Structure of used models.

Model	No. Inputs	No. Hidden Layers	Size of Hidden Layer	No. Outputs	Activation Function
ANN	Variable	3	[100, 50, 100]	1	Linear
RNN	Variable	1	[100]	1	Linear
LSTM	Variable	1	[100]	1	Linear
GRU	Variable	1	[100]	1	Linear

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$TNR = \frac{TN}{TN + FP} \tag{12}$$

$$S_{TTD} = 1 - \frac{1}{n_a}\sum_{i}^{n_a}\frac{TTD_i}{\Delta t_i} \tag{13}$$

$$S = \gamma \cdot S_{TTD} + (1 - \gamma) \cdot S_{CLF} \tag{14}$$

Another important metric when evaluating machine models is the positive predictive value (PPV), which was not considered for the competition. The PPV, also called the precision, is a measure that indicates the proportion of flagged instances that were correctly identified as shown in (15). This is also an important measure in the application environment because the proportion of anomalous instances is far smaller than normal instance so the TNR is always likely to be high. The PPV evaluates how accurate the system is every time an anomaly is flagged which is a very important measure to consider. This is because if, for example, the system is only correct 20% when it flags an anomaly it is unlikely to be taken seriously in practice. The PPV is sometimes combined with the TPR into one metric called the F-score as shown in (16). The F-score, which is the harmonic mean of the two metrics, evaluates whether the system can accurately identify most of the anomalous instances in the dataset.

$$PPV = \frac{TP}{TP + FP} \tag{15}$$

$$F = 2\frac{TPR \cdot PPV}{TPR + PPV} \tag{16}$$

C. MODEL STRUCTURE

The used structures for the ANN, RNN, LSTM and GRU and shown in Table 3. As can be seen from the table the number of inputs going into each neural network architecture is varies based on the dependency matrix shown in Table 1. The ANN structure has three hidden layers with the first, second and third hidden layers having sizes 100, 50 and 100 respectively. Each of the RNN models only has one hidden layer that consists of 100 nodes. All four models have only one output which is the predicted tank level and thus uses the linear activation function. As can be seen in Section II-C, the convolutional neural network has a slightly more complex structure and will thus be discussed separately.

The chosen model for the CNN loosely follows the architecture shown in Figure 3 with some minor structural changes. The model consists of a variable size input layer, six hidden layers, and an output layer corresponding to the predicted tank level. The first two hidden layers are identical

TABLE 4. Neural network results on BATADAL cross-validation dataset.

Rank	Name	No. Attacks.	S	S_{TTD}	S_{CLF}	F_1	TPR	TNR	PPV
1	Ensemble	7	0.9262	0.9463	0.9062	0.7075	0.9002	0.9121	0.5827
2	CNN	7	0.9213	0.9679	0.8748	0.6677	0.8423	0.9072	0.5531
3	LSTM	7	0.8872	0.8971	0.8774	0.7234	0.8144	0.9404	0.6507
4	RNN	7	0.8754	0.8662	0.8847	0.7675	0.8104	0.9589	0.7289
5	GRU	7	0.8721	0.8915	0.8526	0.5977	0.8363	0.8689	0.4650
6	ANN	7	0.8431	0.9024	0.7838	0.5219	0.7026	0.8651	0.4151
7	Naive	7	0.7500	1.0000	0.5000	0.2142	1.0000	0.0000	0.1199

convolutional layers each with a kernel size of three. This is followed by a dropout layer that halves the sized of the network to reduce the probability of overfitting due to the two convolutional layers. The fourth hidden layer performs the max-pooling operation with a pool size of two. The next layer is the flattening layer needed to transform the data into an input that can be used in the last hidden layer, which is a vanilla neural network layer consisting of 100 nodes.

D. GRADIENT OPTIMISATION

Backpropagation happens through a process of gradient optimisation in which the goal is to minimise the objective function [14]. There are many gradient descent optimisation algorithms in use each with its own strengths and weaknesses. In this work the popular adaptive moment estimation (Adam) optimiser [28] is used due to its fast convergence and computational efficiency. Adam is a first-order stochastic optimisation method which uses an adaptive learning rate and combines the popular adaptive gradient and moment-based techniques. The result is a technique that outperforms most other gradient descent algorithms and is consequently the most widely used in practice. Using this algorithm, the network weights are updated using (17) where $\hat{m}_k$ and $\hat{v}_k$ are bias-corrected moment estimates, α is the learning rate, and ϵ is a hyper-parameter.

$$W_k = W_{k-1} - \alpha \frac{\hat{m}_k}{\sqrt{\hat{v}_k} + \epsilon} \tag{17}$$

V. RESULTS

In this sections the results of the neural network architectures presented in the previous sections are evaluated using the BATADAL dataset [27]. The algorithms were trained using only the normal data and the remaining two datasets were used for cross-validation and testing. The results of the former are presented first and then the latter is used to determine how well the algorithms generalised. The performance metrics from the competition will be used to evaluate the algorithms. The algorithms will also be compared to each other and then a multi-stage technique using a univariate anomaly detection will be applied in order to improve results.

A. CROSS-VALIDATION DATASET

The optimal threshold found using the Mahalanobis distance rule is found using the cross-validation dataset. The neural networks were only trained using the normal dataset which means that the cross-validation dataset is also a useful tool in evaluating how well the system works. This is because the dataset is also unseen by the core algorithm and thus the neural network will not know the difference between the cross-validation and test dataset. It is thus possible for an algorithm to be more effective at detecting the attacks in the test dataset than those found in the cross-validation dataset.

The results of running the algorithms on the cross-validation dataset are shown in Table 4. The Naive classifier was not an implemented algorithm but rather a benchmark system assumes all of the data instances are anomalies. This is because the main performance metrics from the BATADAL competition have a bias towards detecting attacks (10)-(14) so any proposed algorithm has to beat the S score of the Naive classifier. The S_{CLF} is the accuracy metric and the S_{TTD} is the time it takes to detect the attack after it starts. All of the neural network architectures are able to beat the Naive classifier by a significant margin in this dataset. In general though the F_1 scores are significantly lower than the S_{CLF} values which is an indication that the algorithms were trained with a bias towards detecting attacks as per the priorities of the application scenario. This approach inherently means that the number of false positives will be high because the threshold attempts to encompass as many anomalies as possible. Training the algorithms to favour the F_1 score instead would however result in fewer attacks being detected which is not ideal so only the BATADAL metrics were considered during training.

Looking at the S scores all of the algorithms were above the 0.8 with CNN and the Ensemble technique both breaching the coveted 0.9 threshold. This can be seen in Figure 6 which plots the S_{CLF} vs the S_{TTD} from the different algorithms. Even though both CNN and the Ensemble technique have S scores above 0.9 only the Ensemble technique has both the S_{TTD} and S_{CLF} above 0.9. Most of the other algorithms have comparable S_{CLF} values with the exception of ANN which is significantly lower than the rest. For ANN the S_{CLF} is below 0.8 but its S_{TTD} is high enough to keep the S also above 0.8. In fact ANN's S_{TTD} is only lower than the two algorithms which are above the 0.9 S score line. This means that ANN detects attacks quickly but is generally less efficient at detecting all of the anomalous instances. This is evident from the fact that it has the lowest number of true positives and the highest number of false positives of any of the algorithms.

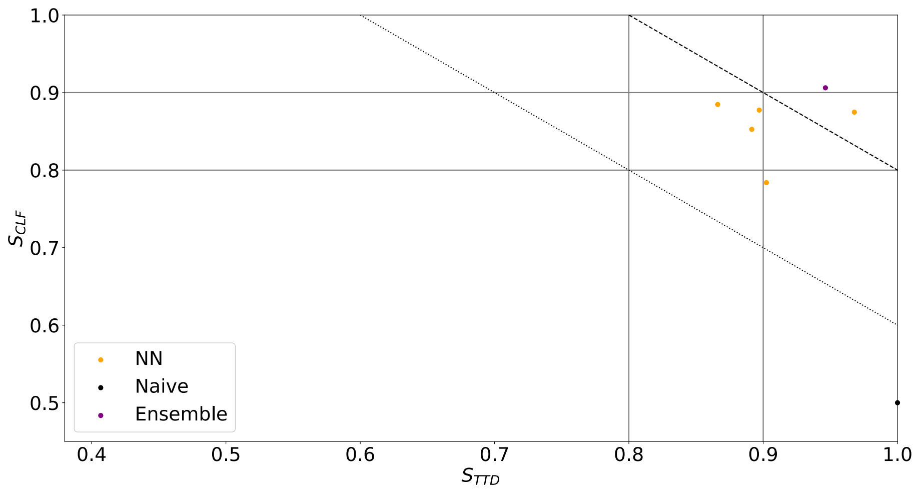


FIGURE 6. NN cross-validation results plotted as a function of S_{CLF} and S_{TTD}.

When looking at the performance metrics from the BATADAL competition the algorithms in general performed really well. The F_1 scores however were not as great with all of them falling below 0.8. Why this is the case is evident when looking at the precision and recall which are the two metrics used to calculate the F_1 score. In general the algorithms have a very high recall rate meaning that they are able to detect most of the attacks found in the dataset. The precisions are however very low meaning that the proportion of flagged instances that are incorrectly identified as anomalous is relatively high. When looking at the proportion of anomalous data instances in the dataset however this is not a major issue. Having a proportionally higher number of false positive among instances that make up only twelve per cent of the dataset will not have significant practical implications because it's still relatively a small number of instances being flagged. The cost of not detecting attacks is much higher than the cost of the minor inconvenience of investigating false cases. Ideally however this value should remain above 0.5 otherwise the number of false positives start outnumbering the number of correctly identified instances.

1) PERFORMANCE OF ENSEMBLE TECHNIQUE

The proposed Ensemble technique outperformed all of the other algorithms on this dataset. This is primarily due to its significantly higher TPR meaning that it was able to identify the most number of attacks of all of the algorithms. This is evident when looking at the number of true positives which is the highest of all the implemented algorithms. Even though the algorithm is composed of the different neural network implementations it is possible to have a higher TP because the different algorithms aren't necessarily flagging the same instances. The combination is able to produce a result that is greater than the sum of its parts. The F_1 score is not the greatest in the group but it falls somewhere in the middle at a respectable value. Its PPV is however significantly above the 0.5 threshold which in general means that the algorithm performed quite well.

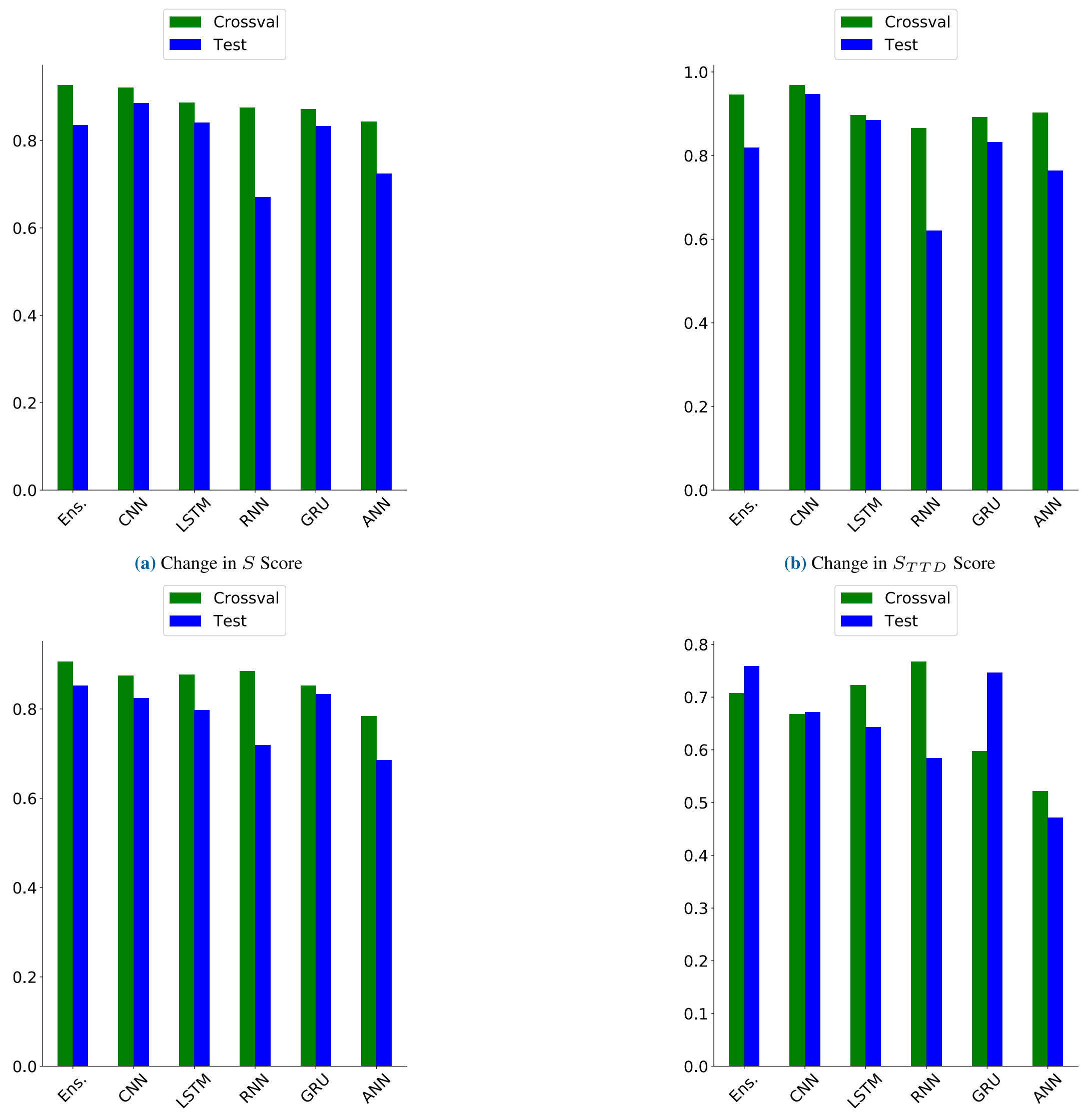
B. TEST DATASET

Once the optimal threshold on the cross-validation dataset is found the test dataset is used to determine how well the algorithms generalise. This dataset is thus completely unseen by both the underlying neural network and the MD algorithm used to find anomalies. The results of this are shown in Table 5 where it is already evident that there is a difference just from the ordering of the algorithms. ANN and RNN perform worse than the naive classifier in this dataset with RNN having the highest reduction in across all performance metrics. It maintains a higher accuracy than ANN when looking at the S_{CLF} and F_1 scores but ANN still has a far superior S_{TTD} making their S scores comparable. RNN was evidently more efficient at detecting outliers in the previous dataset than it was in this one. The remaining algorithms still maintained S score values of above 0.8 but in this dataset none of the algorithms crossed the 0.9 S score mark. The S_{TTD} score of CNN remains above 0.9 but its S_{CLF} score dropped significantly enough for the S score to dip below 0.9.

Figure 7 shows how the performances of the algorithms changed between the key performance metrics. There was a reduction of S scores across the board for all of the algorithms with ANN and RNN being the most affected. The proposed ensemble technique also suffered a heavy loss probably as a result of its S_{TTD} score which dropped significantly on this dataset. The most affected performance metric between the two datasets is the S_{TTD} which had a significant reduction in all of the algorithms except for CNN and LSTM. In fact CNN was the most robust algorithm between the two datasets surpassing the Ensemble technique to become the best performing algorithm in this dataset. The S_{CLF} values also decreased but this difference was not as pronounced as it was for the S_{TTD}. Interestingly enough the F_1 score for half of the algorithm increased in this dataset when compared to the previous one. It is possible for the F_1 score to increase while the remaining performance metrics it is the only one considering precision. While the TPR reduced across the board for the algorithms the PPV of the ones with improved

TABLE 5. Neural network results on BATADAL test dataset.

Rank	Name	No. Attacks.	S	S_{TTD}	S_{CLF}	F_1	TPR	TNR	PPV
1	CNN	7	0.8853	0.9465554	0.8240	0.6716	0.7789	0.8692	0.5903
2	LSTM	7	0.8408	0.8842844	0.7972	0.6432	0.7199	0.8746	0.5813
3	Ensemble	7	0.8355	0.8183917	0.8527	0.7591	0.7666	0.9388	0.7518
4	GRU	7	0.8329	0.8324704	0.8334	0.7471	0.7150	0.9518	0.7823
5	Naive	7	0.7500	1.0000	0.5000	0.3261	1.0000	0.0000	0.1948
6	ANN	6	0.7247	0.7641452	0.6852	0.4721	0.5921	0.7782	0.3925
7	RNN	6	0.6702	0.6208927	0.7196	0.5843	0.4767	0.9625	0.7549



(a) Change in S Score

(b) Change in S_{TTD} Score

(c) Change in S_{CLF} Score

(d) Change in F_1 Score

FIGURE 7. Change in performance metrics between cross-validation and test datasets.

F_1 scored increased. This means that while the algorithms were detecting proportionally fewer attacks, the proportion of relevant flagged instances increased.

1) PERFORMANCE OF ENSEMBLE TECHNIQUE

The performance of the Ensemble technique did indeed reduce significantly between the cross-validation and test datasets as is evident from it dropping two down in the rankings. Looking at the S score in isolation however isn't fully indicative of the performance of the algorithms. This is because the S score was greatly affected by the S_{TTD} score which had one of the largest reduction across the algorithms. Looking at the metrics that evaluate accuracy however the Ensemble technique still has a very strong showing even

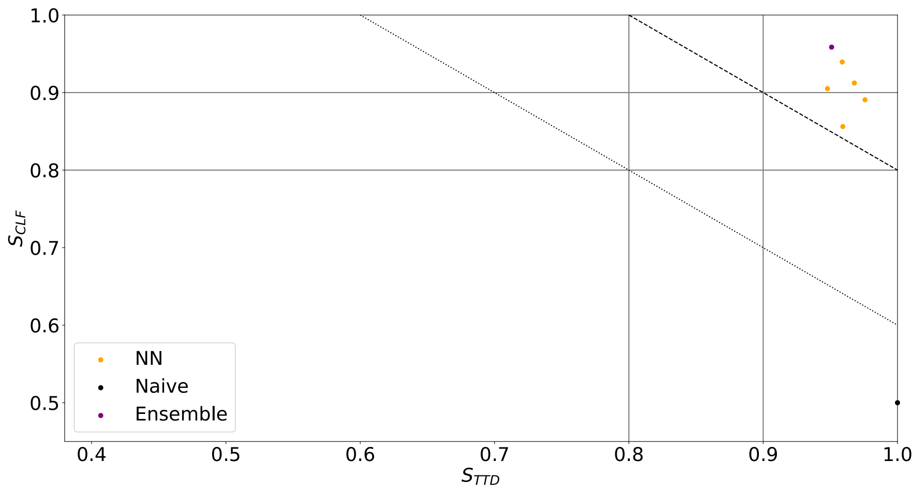


FIGURE 8. Multi-stage cross-validation results as a function of S_{CLF} and S_{TTD}.

in this dataset. The algorithm had the highest S_{CLF} and F_1 scores in this dataset and detected the second most number of attacks behind only CNN. It also had a high precision (close to 0.8) meaning that even though it detected a large number of attacks this wasn't due to the algorithm indiscriminately flagging more instances as anomalous. So in general the technique detected attacks relatively late but was more effective at identifying outliers than the other algorithms.

C. MULTI-STAGE SYSTEM

As mentioned previously we also look at how adding a second stage of detection affects the results. The multivariate anomaly detection algorithm is paired with a univariate detector to create a more accurate multi-stage system. The univariate anomaly detection scheme used is the variation of the boxplot rule which presents a more accurate representation of skewed data and is thus more effective at finding outliers [29]. Figure 8 shows the S_{TTD} plotted against the S_{CLF} for the cross-validation dataset. In the stand-alone systems (Figure 6) only two of the algorithms were above the 0.9 S score line but for the multi-stage systems all of them have crossed it. All of the S_{TTD} scores are also above 0.9 while only two of them have S_{CLF} scores below 0.9. CNN only marginally misses out in this case as its S_{CLF} is less than one percentage point short of the 0.9 mark. The best performing algorithm in this dataset is the Ensemble primarily due to its superior S_{CLF} score. In fact when looking at the metrics that evaluate accuracy the ensemble technique still comfortable outperforms the other algorithms for both the S_{CLF} and F_1 scores. In general though all of the algorithms perform exceedingly well on this dataset with very little difference overall between the top four.

Figure 9 shows the change in performance metrics between the standalone and multi-stage systems in the test dataset. There were significant improvements across all the performance metrics for all the algorithms with CNN remaining the top performing algorithm with the best S score attributed largely to its S_{TTD} which is almost one. GRU also becomes the algorithm with highest S_{CLF} score marginally beating out the Ensemble technique and it also has the second largest F_1 score. The biggest S_{TTD} improvement came from RNN with an almost 20 percentage point increase but its accuracy metrics did not improve enough for it to surpass ANN in the rankings. Both algorithms showed significant improvements but ANN continues to outperform RNN across most performance metrics. This is in line with the rest of the improvements where the ranking remained largely unchanged with the exception of the Ensemble technique which moved down one spot. The Ensemble technique is however far more accurate than the other algorithms and is only in the middle of the rankings because if its lower S_{TTD} score. In general there were significant improvements across all performance metrics in both datasets.

D. COMPARISON WITH BATADAL ALGORITHMS

One of the benefits of using the BATADAL dataset is that there is already a set of machine learning algorithms that were implemented on the same dataset and compared with the same performance metrics. This makes it easier to compare proposed techniques with the state-of-the-art algorithms developed for the competition. Figure 10 shows the results of the implemented neural network architectures compared to the machine learning algorithms proposed in BATADAL implemented on the test dataset. As can be seen from the figure, the implemented architectures have comparable results to the schemes proposed for BATADAL with the top performing algorithms all being within four percentage points of each other. The majority of the algorithms from both sets are either close to or above the 0.9 S score mark with only three falling below 0.8. This shows that the algorithms implemented in the paper perform well when looking at the individual performance metrics and also when comparing them to the state-of-the-art.

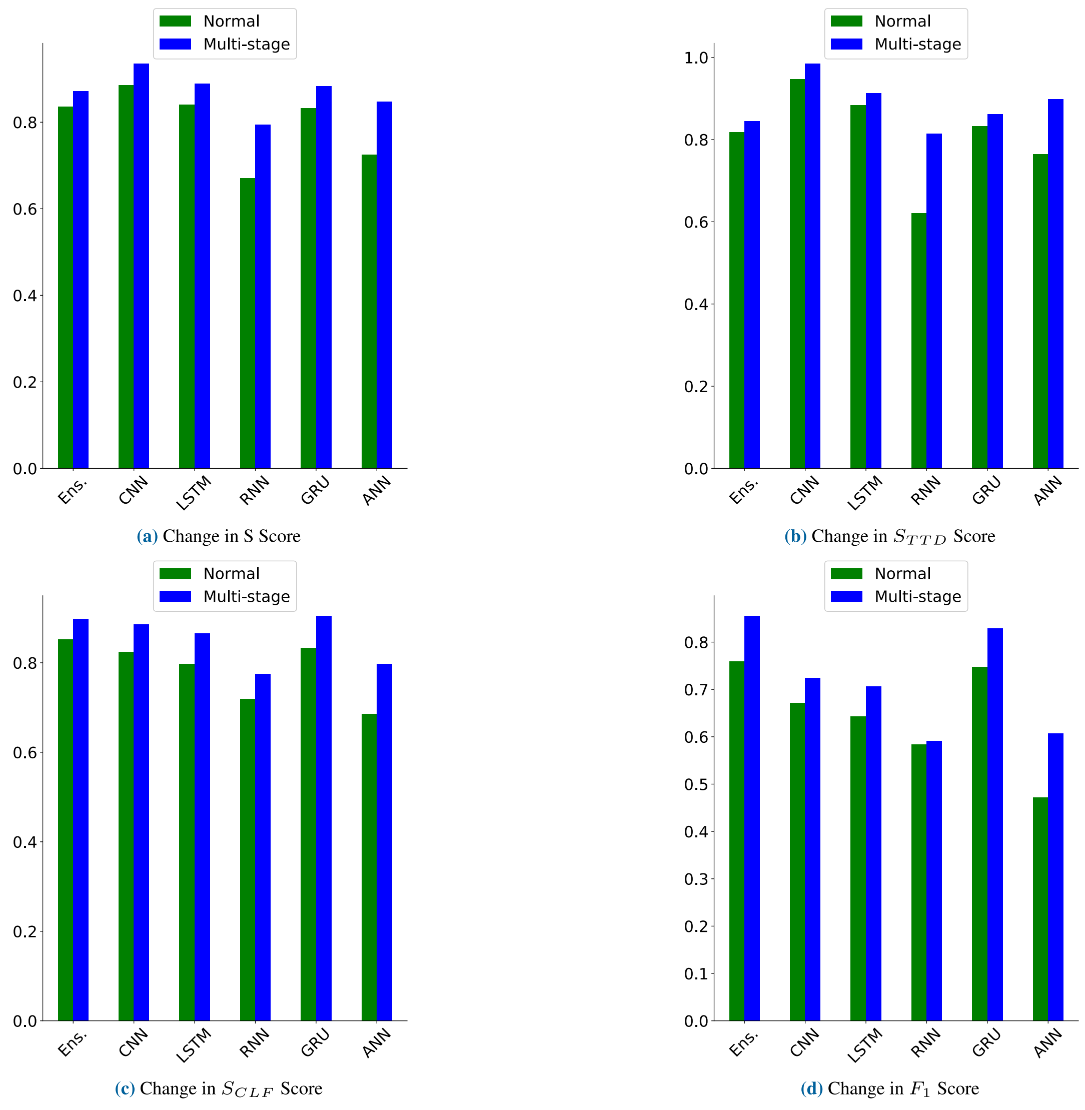


FIGURE 9. Change in performance metrics between normal and multi-stage systems.

VI. OBSERVATIONS

A. NEURAL NETWORKS

In general, all of the algorithms performed reasonably well on the cross-validation dataset but the results were mixed in the test dataset. RNN and ANN were particularly unable to make the transition as they both performed worse than the naive classifier in the test dataset. This means that they aren't able to generalise well which means they are generally able to perform reasonably in the dataset in which the threshold is chosen. GRU and LSTM have comparable results with LSTM slightly taking the edge in most metrics across both datasets. Both algorithms had very good results on both datasets with both maintaining their S and F_1 scores at least 0.8 and 0.6 respectively. The stand out performer across all the algorithms was CNN which was the most robust algorithms between the datasets. It outperforms the other algorithms significantly and maintains the highest TPR across both datasets. This means that it is more effective at finding anomalies than the other algorithms and is also the fastest at detecting them when looking at the S_{TTD} scores. It also maintains an F_1 score of above 0.65 across both datasets which means that it isn't just indiscriminately detecting anomalies like the naive classifier.

B. ENSEMBLE TECHNIQUE

A voting-based ensemble technique was used to combined the results of all of the implemented algorithms. When looking at the S score it had great results in the cross-validation dataset, where it was the best performing algorithm. This is primarily due to its significantly higher TPR meaning that it was able to identify the most number of attacks of all of the algorithms. The combination was thus able to produce a result that is greater than the sum of its parts. The Ensemble technique had mixed results test dataset primarily because

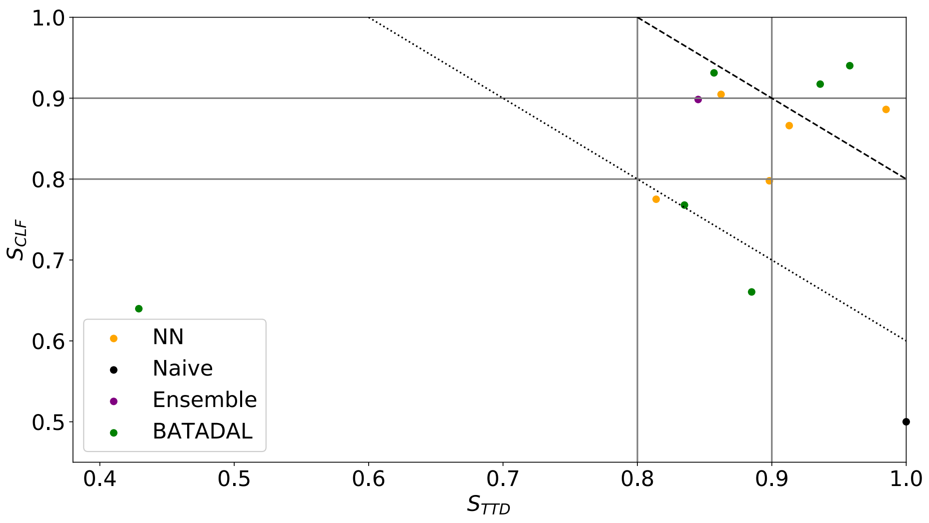


FIGURE 10. Results compared to BATADAL as a function of S_{CLF} and S_{TTD}.

the different architectures also had varying performances. Its S score in the test dataset was largely affected by the fact that it had the lowest S_{TTD} of the top four algorithms. Its greatest strength however was in the performance metrics that evaluated accuracy where it had the highest S_{CLF} across both datasets and maintained an F_1 score of above 0.7. This means the like CNN it is very effective at detecting anomalies but it does so more accurately. This means that even in the test dataset the combination produced a result greater than the individual algorithms.

C. MULTI-STAGE SYSTEMS

Including a multi-stage system increased results considerable across all performance metrics for all of the algorithms. The strength of this approach lies in selecting algorithms that are likely to detect different types of anomalies even if one of the algorithms has a significantly lower accuracy. The is because if the algorithms are detecting the same attacks there will be an overlap in the identified outliers which means that nothing new will be learned from the combination. Combining multiple algorithms has the potential to improve the results but there is also a danger of exacerbating the inadequacies of the combined algorithms. For instance, combining two algorithms could result in an F_1 score that is significantly lower if the combination results in a PPV that proportionally decreases at a higher rate than the TPR increases. This means that considerable thought needs to be put into selecting the different stages of multi-stage systems to ensure the output is more accurate than the individual algorithms.

VII. CONCLUSION

Recurrent neural networks are ideally suited for time-series data because they predict future states using both the current input and previous systems states. This memory of previous system states provides contextual information that improves the accuracy of time-series data prediction. Another popular deep learning architecture is the convolutional neural network which is mostly used in image processing applications. The one-dimensional CNN architecture has however also been found to be very effective when used on time-series data.

In this paper a number of neural network architectures where trained on the normal BATADAL dataset. The dataset consists of fourteen sophisticated attacks that include measures to evade attack detection algorithms. To detect anomalies the output of the networks when run on cross-validation dataset were used to find an error threshold using the multivariate MD rule. A voting-based ensemble technique which combines the outputs of the different algorithms was also implemented to improve results. In general the algorithms performed very well on the cross-validation dataset with slight performance decreases in the test dataset. The only exception was RNN which did not perform well when run on the test data.

The algorithms were compared using a vast range of performance metrics with the S score defined for the BATADAL competition still used as the primary ranking metric. The best performing algorithm between the two datasets when considering the S score was CNN which was very robust across all performance metrics. When looking at the metrics that evaluate accuracy the Ensemble technique was the best performing algorithm across the two datasets. A multi-stage approach combining both the multivariate and univariate outlier detection schemes was also proposed and resulted in a significant improvement across all performance metrics. The models were trained only on the normal data meaning that only a change to the system dynamics would drastically affect the performances of the presented algorithms. The range of attacks used and the fact that they included evasive techniques also means that the results are a good indication of how the algorithms would generalise in a larger dataset encompassing a larger variety of attacks. Future work will evaluate this generalisation in larger more complex industrial control system configurations.

REFERENCES

[1] K. M. Shahanas and P. B. Sivakumar, "Framework for a smart water management system in the context of smart city initiatives in india," *Procedia Comput. Sci.*, vol. 92, pp. 142–147, Jan. 2016.
[2] Y. Chen and D. Han, "Water quality monitoring in smart city: A pilot project," *Autom. Construct.*, vol. 89, pp. 307–316, May 2018.
[3] M. Angelidou, A. Psaltoglou, N. Komninos, C. Kakderi, P. Tsarchopoulos, and A. Panori, "Enhancing sustainable urban development through smart city applications," *J. Sci. Technol. Policy Manage.*, vol. 9, no. 2, pp. 146–169, Jul. 2018.
[4] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 860–880, 2nd Quart., 2013.
[5] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 13, Dec. 2019.
[6] Z. Inayat, A. Gani, N. B. Anuar, S. Anwar, and M. K. Khan, "Cloud-based intrusion detection and response system: Open research issues, and solutions," *Arabian J. Sci. Eng.*, vol. 42, no. 2, pp. 399–423, Feb. 2017.
[7] S. Axelsson and D. Sands, *Understanding Intrusion Detection Through Visualization*. Basel, Switzerland: Springer, 2006.
[8] A. Patel, M. Taghavi, K. Bakhtiyari, and J. C. JúNior, "An intrusion detection and prevention system in cloud computing: A systematic review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 25–41, 2013.
[9] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, p. 3, Dec. 2018.
[10] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, Aug. 2018.
[11] C. Aggarwal, "An introduction to outlier analysis," in *Outlier Analysis*. Basel, Switzerland: Springer, 2017, pp. 1–34.
[12] G. Shmueli, P. C. Bruce, I. Yahav, N. R. Patel, and K. C. Lichtendahl, Jr., *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*. Hoboken, NJ, USA: Wiley, 2017.
[13] D. Kriesel, "Components of artificial neural networks," in *A Brief Introduction on Neural Networks*, 2007, pp. 37–58. [Online]. Available: https://www.dkriesel.com
[14] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*, 1st ed. Basel, Switzerland: Springer, 2017.
[15] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Dept. Inform., Tech. Univ. Munich, Munich, Germany, 2008.
[16] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 816–833.
[17] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN Encoder–Decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
[18] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.
[19] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 664–675, Mar. 2016.
[20] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3367–3375.
[21] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8614–8618.
[22] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, pp. 255–258.
[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[24] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," in *Proc. Int. Workshorp Detection Classification Acoustic Scenes Events (DCASE)*, 2017, pp. 80–84.c.
[25] C. Leys, O. Klein, Y. Dominicy, and C. Ley, "Detecting multivariate outliers: Use a robust variant of the mahalanobis distance," *J. Experim. Social Psychol.*, vol. 74, pp. 150–156, Jan. 2018.
[26] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher, "FastDeepIoT: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proc. 16th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2018, pp. 278–291.
[27] R. Taormina, S. Galelli, N. O. Tippenhauer, E. Salomons, A. Ostfeld, D. G. Eliades, M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks, and B. M. Brentan, "Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *J. Water Resour. Planning Manage.*, vol. 144, no. 8, 2018, Art. no. 04018048.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[29] M. Hubert and E. Vandervieren, "An adjusted boxplot for skewed distributions," *Comput. Statist. Data Anal.*, vol. 52, no. 12, pp. 5186–5201, Aug. 2008.

TSOTSOPE DANIEL RAMOTSOELA (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in computer engineering from the University of Pretoria, in 2013, 2015, and 2020, respectively. He is currently a Lecturer with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria. His research interests include system security, machine learning, and wireless sensor networks, primarily focusing on IoT applications and cyber-physical systems.

GERHARD PETRUS HANCKE (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Pretoria, South Africa, in 2002 and 2003, respectively, and the Ph.D. degree in computer science from the Computer Laboratory, Security Group, University of Cambridge, in 2009. He is currently an Associate Professor with the City University of Hong Kong, Hong Kong, SAR. His research interests include system security, embedded platforms, and distributed sensing applications related to the industrial Internet-of-Things.

ADNAN M. ABU-MAHFOUZ (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees in computer engineering from the University of Pretoria. He is currently the Centre Manager of the Emerging Digital Technologies for 4IR (EDT4IR) Research Centre, Council for Scientific and Industrial Research (CSIR), a Professor Extraordinaire with the Tshwane University of Technology, a Visiting Professor with the University of Johannesburg, and an Extraordinary Faculty Member with the University of Pretoria. He is the Founder of the Smart Networks collaboration initiative that aims to develop efficient and secure networks for the future smart systems, such as smart cities, smart grid, and smart water grid. He participated in the formulation of many large and multidisciplinary research and development successful proposals (as a principal investigator or main author/contributor). His research interests include wireless sensor and actuator networks, low power wide area networks, software defined wireless sensor networks, cognitive radio, network security, network management, and sensor/actuator node development. He is a member of many IEEE Technical Communities. He is an Associate Editor of IEEE Access, the IEEE Internet of Things, and the IEEE Transactions on Industrial Informatics.