Technical Report

RISE Germany Internship: Applying Deep Learning Methods to the Search for Astrophysical Tau Neutrinos

William Martin

11/2017

technische universität dortmund

SFB 876 Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung

Part of the work on this technical report has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", project C3.

Speaker: Prof. Dr. Katharina Morik
Address: TU Dortmund University
Joseph-von-Fraunhofer-Str. 23
D-44227 Dortmund
Web: http://sfb876.tu-dortmund.de

1 Introduction

IceCube is a neutrino detector built into the ice at the South Pole [1]. It has a volume of approximately one cubic kilometre, and consists of 86 strings of 60 Digital Optical Modules (DOMs) each, embedded between 1450 and 2450 metres below the surface. Each DOM is composed of a photomultiplier tube connected to two different analogue-to-digital converters that digitise the output of the photomultiplier tube over two different timescales. Of particular interest to this work is the Analogue Transient Waveform Digitiser (ATWD), which takes 128 samples with 3.3ns spacing. Since the detector's completion in 2010, an astrophysical neutrino flux has been measured [2] and several high-energy astrophysical neutrinos have been observed [3].

There are two main event topologies for neutrino interactions [4]. The first is made up of "track-like" events, which occur when a charged current muon neutrino interaction produces a muon; the muon propagates through the ice and out of the detector, producing a track of Cherenkov radiation. The second type of event is a "cascade-like" event, which arises from charged current interactions of electron and tau neutrinos, and neutral current interactions of all flavours of neutrino. These interactions are characterised by a flood of particles, dumping their energy over a relatively small area of the detector.

Thus far, observations of astrophysical tau neutrinos have remained elusive [5]. Charged current tau neutrino interactions have two interaction vertices: at the first, a tau particle is produced; this tau particle propagates through the detector before decaying to produce the second vertex. This event profile leads to two cascades, one at each vertex. At low energies, the two vertices are very closely spaced and are therefore indistinguishable from other "cascade-like" events. However, at higher energies (>100 TeV), these two interaction vertices are separated enough such that the two vertices can be resolved, allowing these CC ν_τ events to be distinguished from other events — such events are known as "double-bang" events. This work is concerned with "double-pulse" events, a subset of "double-bang" events where the interaction vertices are spaced close-enough together that both interaction vertices are visible from a single DOM, but still far enough apart to be resolved from simple cascade events. These events are exceedingly rare, therefore identifying them in amongst the background of all other neutrino interactions and atmospheric muons poses a technical challenge.

Previous work on identifying double-pulse events has relied upon a series of hard cuts on waveforms measured by the ATWD [5]; an improved analysis was later carried out using machine-learning techniques [6]. This analysis applies results from the field of deep learning to this identification problem, exploring the efficacy of three different neural network types: convolutional neural networks, self-normalising neural networks and recurrent neural networks. Convolutional neural networks [7, 8], currently favoured for image classification tasks, are well-suited to identifying features and patterns in data independent of its position within that data. Self-normalising neural networks [9] are a new form of feed-forward neural network that have shown promise in astronomical classification tasks. Recurrent neural networks [10], and in particular Long Short-Term Memory networks (LSTMs) [11], are often applied to datasets involving time series or sequential data, excelling at tasks such as language processing and speech recognition.

The following section of this work discusses the methods used to create and train the neural networks; Section 3 characterises the performance of these networks. Section 4 presents a brief discussion of the results attained, before ending with conclusions drawn.

2 Methods

This section begins with a discussion of the training data, including the preprocessing performed, followed by the different neural network architectures used, and concluding with the training regime employed.

2.1 Training Data

For the type of neural networks used, it is necessary to have a labelled set of data with which to train the network. The data consisted of simulated neutrino interactions taken from Level 2 of the IceCube data processing pipeline. The background of atmospheric muons had been ignored, as these can be excluded later using other processing methods. For the same reason, the background of CC ν_μ interactions had also been excluded. A series of generous cuts had then been applied on the geometry of the CC ν_τ interactions and the derivatives of their resultant waveforms — these waveforms were labelled as "double-pulse waveforms", and the remainder of the CC ν_τ interactions were discarded. This ensured that only unambiguously "double-pulse-like" waveforms were labelled as such. The waveforms from CC ν_e interactions and NC interactions of all flavours had been labelled as "cascade-like background". A major drawback of this pre-preparation method was that there was a severe imbalance between the "double-pulse waveforms" ($\sim$0.2%) and "cascade-like background" ($\sim$99.8%) — methods for combating the effects of this are discussed in Section 2.3.

As is often done in machine learning, this dataset was split into three: a training dataset (80%), used to train each network; a validation dataset (7%), used for assessing the performance of the networks both during and after training; and a test dataset (13%), which was only used to quantify the performance of the best network (selected on the basis of its performance on the validation dataset). The data was then preprocessed by calculating the transformation that would give the training dataset a mean of 0 and a variance of 1, and applying it to each dataset.

2.2 Neural Network Architecture

Three different types of neural network were investigated. Particulars of each architecture and hyperparameters were largely left untouched once set, as the focus was on comparing the performance of different networks, not on optimising the performance of one type of network. All networks were used with a categorical cross-entropy loss function.

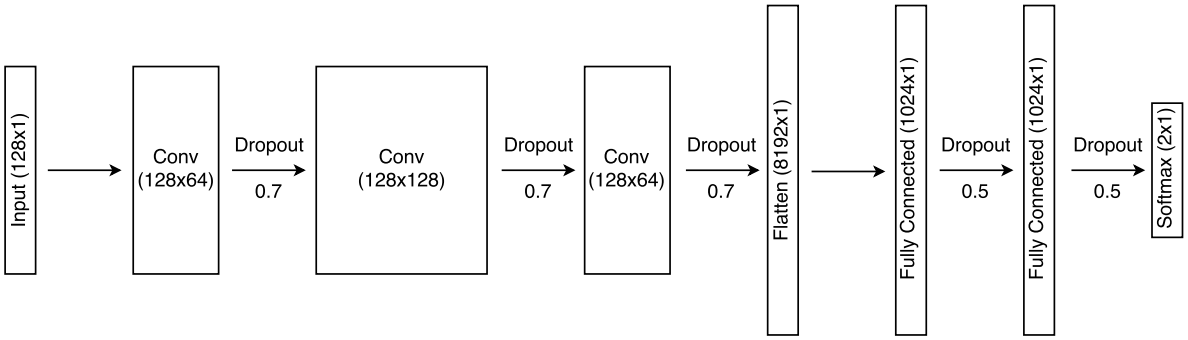


Figure 1: The structure of the convolutional neural network, with dimensions of each layer marked in brackets. Where dropout was used between layers, the arrows have been annotated to indicate the fraction of inputs dropped. The first convolutional layer used a filter size of 5; subsequent convolutional layers used a filter size of 3. All convolutional layers used padding such that the size of the output was preserved.

2.2.1 Convolutional Neural Networks (CNNs)

A convolutional neural network [7, 8] was designed to identify the double-pulse waveforms in our dataset; the structure used is noted in Figure 1. ReLU neurons were used throughout the network. Dropout [12] was used after each layer to reduce overfitting and increase the robustness of the network. The hyperparameters for the CNN (namely the architecture used, the filter sizes of each convolutional layer, and the amount of dropout used) were not optimised.

2.2.2 Self-Normalising Neural Networks (SNNs)

A self-normalising neural network [9] was also applied to this classification problem — the exact architecture is illustrated in Figure 2. Alpha dropout was used between layers, as suggested in Klambauer et al. [9]. Once again, hyperparameters were not optimised.

2.2.3 Long Short-Term Memory (LSTM)

The final neural network structure tried was an LSTM [11]. Three layers of 128 LSTM units were stacked; ReLU neurons were used throughout, and dropout was applied both to the input to each LSTM and to its hidden state vector with a drop probability of 0.2. The final layer was a 2-neuron softmax classification layer. As before, no hyperparameter optimisation was performed.

2.3 Training

Each network was trained on the training data over 10000 epochs; each epoch consisted of 100 batches of 128 samples, with each batch drawn from a binomial distribution of the two classes with expected probability p_{DP} for a double-pulse waveform. This was to

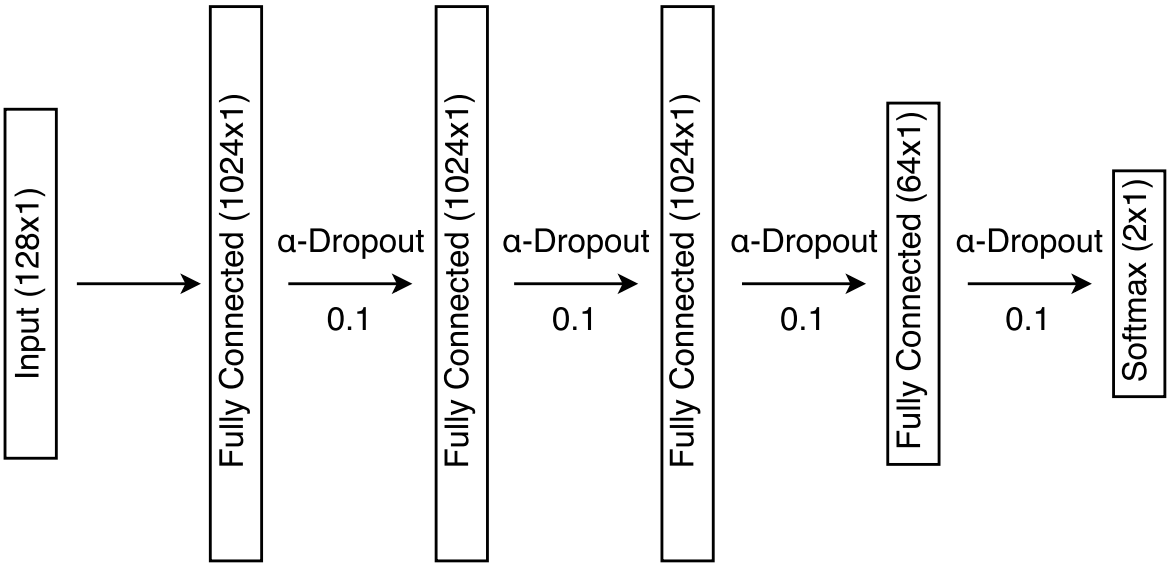


Figure 2: The structure of the self-normalising neural network, with dimensions of each layer marked in brackets. Alpha dropout was used between layers; the arrows between layers have been annotated to indicate the fraction of inputs dropped.

counteract the class imbalance in the original dataset – when trained with the original class balance of 0.2% double-pulse waveforms, the networks tended to predict that all samples were cascade-like background.

While training each network, the network's accuracy on the training batches was monitored, as was its accuracy on batches of validation data drawn in the same way as the training data. Overfitting to the training data was observed when these two accuracies were compared. This problem was fixed using L2 regularisation applied to the weights of each layer [13], which adds a term to the loss function weighted by a multiplicative pre-factor of α. Different levels of regularisation ($\alpha = 0$, 10^{-2}, 10^{-3} and 10^{-4}) were tested on the convolutional neural network. Without regularisation ($\alpha = 0$), heavy overfitting was observed; this was partially ameliorated by using $\alpha = 10^{-4}$, but overfitting was still observed. When $\alpha = 10^{-2}$ was used, no overfitting was observed but the performance of the neural network on the data was found to deteriorate. A regularisation of 10^{-3} was found to strike an appropriate balance between preventing overfitting and having a detrimental impact on the neural network performance.

Another method trialled for improving the neural network performance was decaying p_{DP} after each epoch (henceforth referred to as "balance decay"), using the relationship

$$p_{\mathrm{DP}}(n) = \frac{p_0}{(1+\beta n)^n} \tag{1}$$

where n is the epoch number, p_0 is the initial probability of a double-pulse waveform, and β is a constant parameter that sets the rate of decay. A CNN was trained from scratch using this method; it was also used with a CNN that had been previously trained without using balance decay, after which all but the final fully-connected layer and the softmax layer were frozen, and the network trained for a further 5000 epochs using balance decay.

3 Results

The network performances were evaluated on the validation dataset. The results of this evaluation were used to pick a best model, which was then evaluated on the test dataset.

Purity-efficiency plots for each network were generated using the validation dataset — a selection is shown in Figure 3. In addition, the ν_τ CC events that had previously been excluded were reintroduced, and a histogram was made of expected event rate (for ν_τ CC events and cascade-like background) for a given confidence level — some of the plots made can be seen in Figure 4.

Interestingly, it was found that, despite the overfitting exhibited in the CNN that did not employ regularisation and better train-time accuracy on the validation data when balanced batches were fed into the system for the best-performing regularised CNN (with $\alpha = 10^{-3}$), the unregularised CNN performed better on the validation data. This unregularised CNN (CNN-A) provided the basis for a CNN that was trained using the balance decay method as described in 2.3; the resultant network (CNN-B) proved to have the best performance of any of the networks.

The performance of SNNs was consistently worse than either of the other network architectures (as may be expected, given that SNNs do not account for the continuous nature of the waveform data). LSTMs showed promise, but ultimately performed worse than the best-performing CNNs.

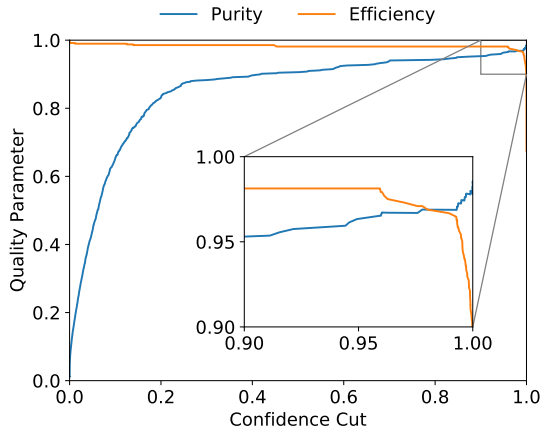
On the basis of the plots shown in Figures 3 and 4, CNN-B was chosen as the best-performing network. The network was then evaluated on the test dataset (using all ν_τ CC events), as well as on the previously-discarded ν_μ CC events and atmospheric muons (using a dataset previously generated with CORSIKA [6]). The expected number of events per year at a given confidence threshold were plotted both for the test dataset alone and for the combined dataset including the ν_μ CC events and the atmospheric muons – these plots are illustrated in Figure 5.

While a confidence cut has not been optimised and chosen, it can be seen that if a cut of, for instance, 0.62 were used, it would be expected that one ν_τ CC event would be detected every 1.5 years, with a rate of one cascade-like background event every 83 years. The rate of ν_μ CC events is approximately a factor of 3-4 higher than that of ν_τ events; the rate of atmospheric muon events is approximately four orders of magnitude higher.

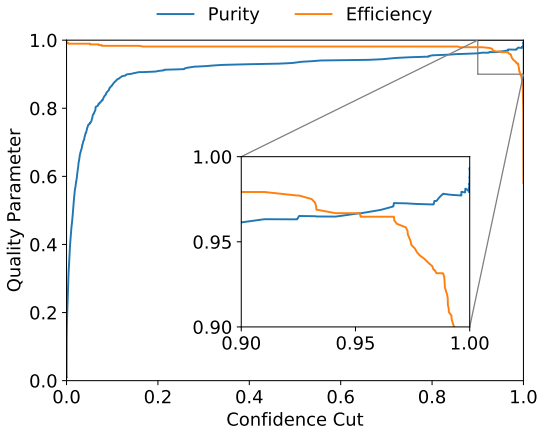
4 Discussion

While further steps need to be taken to remove ν_μ CC events and atmospheric muon events, these results compare favourably with previous work done. Aartsen et al. [5] uses a system of hard cuts on the derivatives of the waveforms, and expects one ν_τ CC event every 4.6 years, with a background level of ν_e events of one every 42 years. The machine learning approach described in Aartsen et al. [6] reports similar expectations to the analysis presented here when compared with the same astrophysical flux assumption.

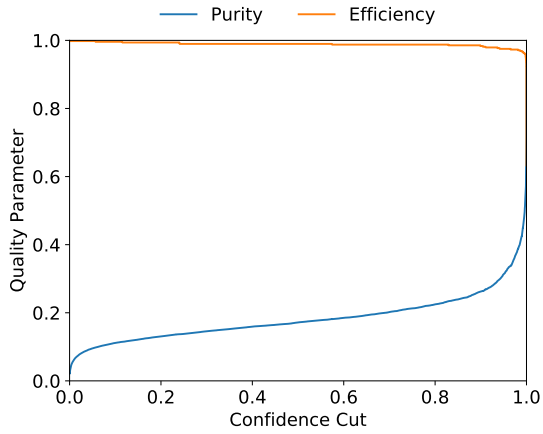
It must be noted that further steps must be taken to remove atmospheric muon and



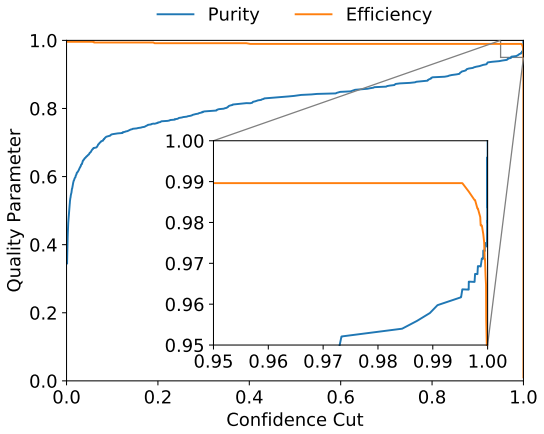
(a) CNN-A (no regularisation)



(b) CNN-B (no regularisation, with balance decay)

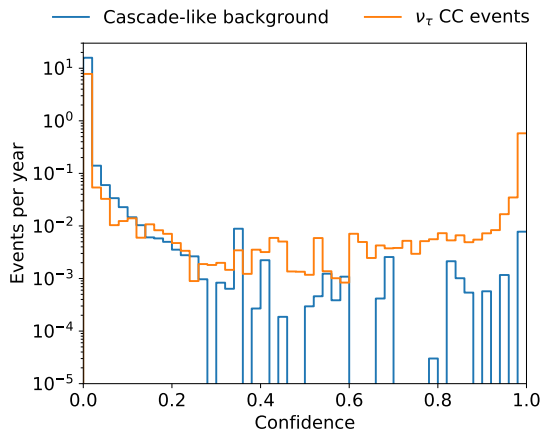


(c) SNN

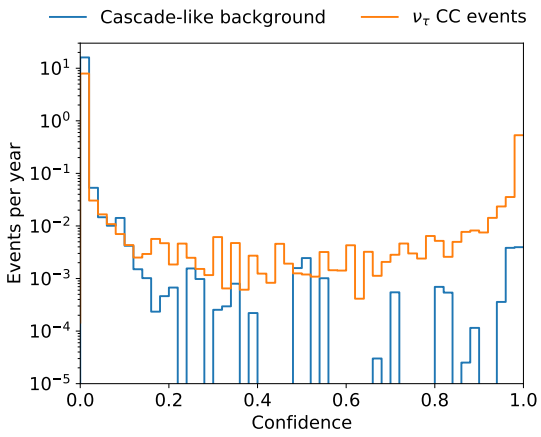


(d) LSTM

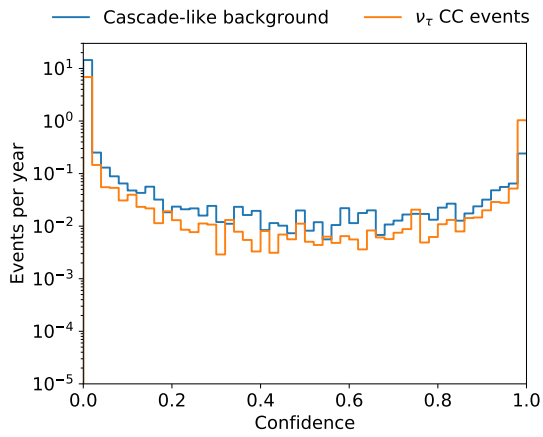
Figure 3: Purity-efficiency plots for the two best-performing CNNs and the best-performing SNN and LSTM. The performance of CNN-A and CNN-B are comparable. The LSTM performs better than the SNN but not as well as either CNN-A or CNN-B. The SNN is the worst-performing network.



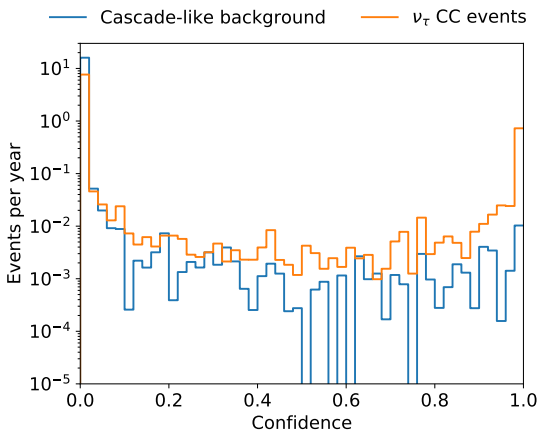
(a) CNN-A (no regularisation)



(b) CNN-B (no regularisation, with balance decay)

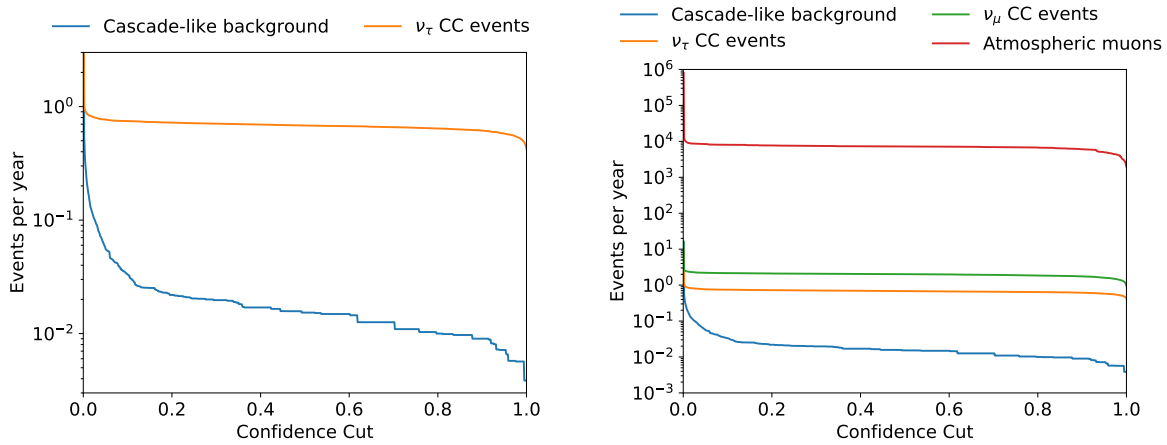


(c) SNN



(d) LSTM

Figure 4: Event-rate plots for the two best-performing CNNs and the best-performing SNN and LSTM. In this figure, the performance of CNN-A and CNN-B can be distinguished. As in Figure 3, both outperform the LSTM, which in turn outperforms the SNN.



(a) Test dataset

(b) Extended dataset

Figure 5: Plots illustrating expected event rate for a given confidence cut, as calculated for CNN-B on the test data. For a confidence cut of 0.62, it would be expected that there would be one ν_τ CC event every 1.5 years, with a background rate of one cascade-like event every 83 years. It can also be seen that, using this neural network alone, the rate of ν_μ CC events passing this cut would be a factor of 2-3 higher than that of ν_τ CC events, and the rate of atmospheric muons would be approximately four orders of magnitude higher.

muon neutrino events, which contaminate the final performance of the network. Once these steps have been taken, it would be possible to derive a confidence cut quantitatively. Furthermore, no errors have been derived on any of the results because of time constraints; it would be necessary to place boundaries on some of the values quoted to allow meaningful comparison. One way of adding errors could include the training of multiple networks of the same architecture in the same way, and quantifying the spread of their performance on the data.

In addition, this work has proved the efficacy of neural networks when applied to this problem, but has not performed detailed optimisation either of the structure of the network or of hyperparameters used during training. In particular, because of the lengthy training time for the LSTMs, very little in the way of optimisation was able to be done. It is probable that the performance of all networks could be improved if further time were available to carry out this optimisation.

Once these improvements have been made, the performance on actual data can be examined, and an attempt at finding potential candidate ν_τ events could be made.

5 Conclusion

The performance of convolutional neural networks, self-normalising neural networks, and long short-term memory networks were assessed for the identification of charged current tau neutrino events from a background of cascade-like events (charged current electron neutrino interactions and neutral current neutrino interactions of all flavours).

Each network was trained using Monte Carlo event simulations. During training, multiple problems were encountered: class imbalance in the training data was resolved by generating balanced batches of event types and employing a balance decay method to progressively decrease the artifical balance generated; overfitting to the training data was overcome by regularisation techniques such as dropout and weight decay.

It was found that CNNs had the best performance on the validation data, with LSTMs performing moderately well, and SNNs being the least effective of all three network types. The performance of the CNN was improved by freezing the top layers of the best network to that point and retraining using the balance decay method; it was this network that had the best performance of any of the networks.

This network's performance on a test dataset of Monte Carlo event simulations was then quantified. For a confidence cut of 0.62, an event rate of one per 1.5 years for ν_τ CC events would be expected. By comparison, the expected cascade-like background event rate was one event every 83 years.

References

[1] A. Achterberg et al. "First year performance of the IceCube neutrino telescope". In: *Astroparticle Physics* 26.3 (2006), pp. 155–173.

[2] M. G. Aartsen et al. "Observation of High-Energy Astrophysical Neutrinos in Three Years of IceCube Data". In: *Phys. Rev. Lett.* 113 (2014), p. 101101.

[3] M. G. Aartsen et al. "First Observation of PeV-Energy Neutrinos with IceCube". In: *Phys. Rev. Lett.* 111 (2 July 2013), p. 021103. DOI: 10.1103/PhysRevLett.111.021103. URL: https://link.aps.org/doi/10.1103/PhysRevLett.111.021103.

[4] Thomas K. Gaisser. *Cosmic rays and particle physics / Thomas K. Gaisser, Ralph Engel, Elisa Resconi.* eng. Second edition. Cambridge University Press: Cambridge University Press, 2016. ISBN: 9781139192194.

[5] M. G. Aartsen et al. "Search for Astrophysical Tau Neutrinos in Three Years of IceCube Data". In: *Physical Review D* 93 (2016), p. 022001.

[6] M. G. Aartsen et al. "The IceCube Neutrino Observatory - Contributions to ICRC 2017 Part II: Properties of the Atmospheric and Astrophysical Neutrino Flux". In: (2017), pp. 86–93. arXiv: 1710.01191 [astro-ph.HE].

[7] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (Winter 1989), pp. 541–551.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[9] Günter Klambauer et al. "Self-Normalizing Neural Networks". In: *CoRR* abs/1706.02515 (2017). URL: http://arxiv.org/abs/1706.02515.

[10] M. C. Mozer. "A Focused Backpropagation Algorithm for Temporal Pattern Recognition". In: *Complex Systems* 3 (1989), pp. 349–381. URL: http://www.complex-systems.com/pdf/03-4-4.pdf.

[11] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[12] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[13] Andrew Y. Ng. "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance". In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: ACM, 2004, pp. 78–. ISBN: 1-58113-838-5. DOI: 10.1145/1015330.1015435. URL: http://doi.acm.org/10.1145/1015330.1015435.

[14] François Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.