Deep Learning Applications in Flavour Tagging

KA WANG KWOK

Deep Learning Applications in Flavour Tagging

Ka Wang Kwok

A Thesis presented for the degree of
Doctor of Philosophy

Institute for Particle Physics Phenomenology
Department of Physics
Durham University
United Kingdom

May 2022

Deep Learning Applications in Flavour Tagging

Ka Wang Kwok

Submitted for the degree of Doctor of Philosophy

May 2022

Abstract: Motivated by the application of data-driven solutions to the field of particle physics, in particular flavour tagging, we study the effectiveness of deep learning (DL) approaches for inclusive $|V_{ub}|$ measurement within the Belle II environment and strangness tagging in the LHCb environment.

In the $|V_{ub}|$ study, we compare the performance of an existing Boosted Decision Tree approach with a Bayesian neural network. In addition, we perform an in depth study on the selected features, investigating the signal inclusivity of DL models which gives insights into behaviours of the models.

We aim for classification speed and precision in the strange-quark jets tagging study. Therefore, we explore using a simple fully connected feedforward neural network to classify s-jets among all light jet backgrounds. A comprehensive feature investigation is performed to understand the discriminating power of jet observable J_s and the importance of particle identification.

Additionally, data-driven methodologies are also reshaping industrial practices. A study investigating the potential of DL in predicting realised volatility of a financial index is included. It is a collaborative project with Optiver where neural networks along with various training schemes are studied to maximise profits.

Contents

List of Figures

List of Tables

Declaration

The work in this thesis is based on research carried out at the Institute for Particle Physics Phenomenology, the Department of Physics, Durham University, England. No part of this thesis has been submitted elsewhere for any degree or qualification. This thesis is based on research which has been carried out in collaboration with Jack Araz, Anke Biekötter and Benjamin D. Pecjak.

- Chapter 4 is based on the article "*Potential and limitations of machine-learning approaches to inclusive* $|V_{ub}|$ *determinations*" [57].

Acknowledgements

First and foremost, I would like to thank my supervisor, Benjamin Pecjak, for his patient supervision and guidance throughout my PhD and without whom this work would not have been possible. I would also like to thank my collaborators: Anke Biekötter and Jack Araz for showing me the ways of academic research. None of this work would have been possible without the funding from STFC and Durham University which I am truly grateful for.

A special mention goes to Frank and Carlton for running the Data Intensive Science with Particle Physics programme which has provided many insights into both academia and industry.

I would like to thank the amazing community at the IPPP, without whom my experience would not have been the same. I will always miss the entertaining lunch time chats and intriguing conversations in the offices. I would also like to thank Alexander Lenz and the rest of the IPPP Outreach Team for the inspiring adventures and opportunities. I am also grateful to Ryan, Henry, Viraj and Anke, who kindly proofread parts of this thesis.

Another special mention goes to Trudy, Linda, Joanne, Adam and Paul for their endless assistance in resolving any admin and IT problems within the IPPP.

I would like to thank Edward, Ingi, Kevin and the rest of the systematic trading research team at Optiver for an inspiring and insightful six months internship into the financial sector.

Last but not least, I would like to thank my mother, Eunice, for the sacrifices she made to send me abroad and endless support throughout my studies.

Chapter 1

Introduction

Machine learning (ML) applications have evolved rapidly for most fields of scientific studies, this includes subjects relevant to particle physics. ML has been applied to various problems, beginning with applications to high-level physics analysis in the 1990s and 2000s, followed by an explosion of applications in particle and event identification and reconstruction in the 2010s [1]. A key objective of particle physics since the major discovery of the Higgs boson has been to capitalise the full physics potential of both the Large Hadron Collider (LHC) and other upcoming experiments such as the high luminosity LHC (HL-LHC) and the B physics focused Belle II. There will be a vast collection of new challenges from such experiments, both quantitatively and qualitatively, as the event size, data volume and complexity reaches new heights. The physics reach from interpreting the experimental results will be limited by efficiencies and performances of algorithms and computational resources. The application of ML to particle physics shed light on both of these areas.

In this thesis, we focus on two event classification problems. The first one is the inclusive measurement of $|V_{ub}|$ in the Belle II environment where $|V_{cb}|$ related background events dominate. We compare the performances of a deep learning based Bayesian neural network (BNN) with the existing boosted decision tree (BDT) analysis. Further investigations include complexity of features, the usage of different Monte Carlo (MC) event generators and the inclusivity of phase space probed by the trained models.

The other problem concentrates on implementation of a strangeness tagger for LHCb as the luminosity upgrades transform it into a general purpose detector. We explore the possibility of having a simple neural network classifying s type jets from other light jet backgrounds. ATLAS and CMS scenarios are briefly explored under the same setup.

The approaches presented for both problems involve supervised learning, as the use of MC event generators remove sample size constraints which is hugely beneficial for data-driven analytics. However, this also means that the dependence on the quality of modelling within the MC is much larger in comparison to other data sources. This is inevitable within particle physics as the purpose of these classifiers is to further our understanding of experimental data. This dependence is investigated in the context of the problems, in particular the $|V_{ub}|$ measurement where an inclusivity study of the models depending on their choice of MC and input features is included.

While ML is popular among scientific studies, it is also transforming other industries. Most sectors have a vast volume of data untouched once recorded. The value of these historical data are being recognised through ML. An additional aspect of this thesis is based on the internship carried out as part of the study, whereby a research project was undertaken in collaboration with Optiver. The purpose of this project was to explore whether deep learning methods can be used to predict realised volatility of a financial index. A selection of neural network architectures, input features and training methods were investigated with a strong focus on the profits and losses generated as the performance metric.

The rest of this thesis is organised as follows. Chapter 2 is an overview of the relevant parts of the Standard Model (SM). Chapter 3 describes the ML algorithms used throughout the studies presented and they include BDT, NN and the metrics used to evaluate their performances. Chapter 4 is a comprehensive study on ML-approaches in $|V_{ub}|$ inclusive measurements. Chapter 5 explores the potential of building a strangeness tagger built from simple NNs. Chapter 6 provides an overview on realised volatility and option theory before diving into the performance investigation from various ML techniques and we will conclude in Chapter 7.

Chapter 2

Physics Background

This thesis explores two event classification problems that are fundamentally flavour physics problems. The central message from both problems is to provide a tool in testing the SM. This chapter focuses on giving an overview of the SM while focusing on the flavour sector. The aim is to provide sufficient understanding for the research results presented in later chapters. The contents are largely based on Ref. [2–4].

2.1 The Standard Model

The current best theory in describing fundamental particle interactions is the Standard Model (SM). It has been extensively tested by experimental results from colliders such as the Large Hadron Collider (LHC) at CERN. The most notable discovery is the observation of a Higgs-like scalar boson in 2012 by ATLAS [5] and CMS [6]. The SM is great at describing a compelling collection of phenomena but there are also several long standing unaddressed problems. To name a few, an explanation for the neutrino masses, particle content for Dark Matter and the existence of Dark Energy. The SM is a quantum field theory (QFT) which describes the fundamental interactions of nature apart from gravity. The descriptions are governed by the Lagrangian density $\mathcal{L}_{\text{SM}}$, typically denoted as just the Lagrangian. It can be expressed by the following schematic equation:

$$\begin{aligned}\mathcal{L}_{\mathrm{SM}} = &-\frac{1}{4}F_{\mu\nu}F^{\mu\nu} \\ &+ i\bar{\psi}\not{D}\psi \\ &+ \bar{\psi}_i y_{ij}\psi_j H + \bar{\psi}_i y_{ij}\psi_j \tilde{H} \\ &+ |D_\mu H|^2 - V(H)\ ,\end{aligned} \tag{2.1.1}$$

where each line respectively represents pure gauge, matter, Yukawa and Higgs. Explicitly, the first line contains kinetic and interaction terms of gauge fields stored within the field strength tensor $F^{\mu\nu}$. The second line describes the propagation of matter fields ψ and their interactions with gauge fields through the covariant derivative $\not{D}$. The third line is known as the Yukawa interaction where y is the Yukawa matrix. This line describes the interaction between the Higgs field H and matter fields, giving rise to their masses after electroweak spontaneous symmetry breaking (EWSB). The final line represents the Higgs-field interactions with the gauge fields and with itself. The self-interaction produces the potential. More details on matter field masses are explained in Section 2.1.2.

The SM is a gauge theory constructed with the $SU(3)_c \times SU(2)_L \times U(1)_Y$ gauge symmetry where the subscripts c represents the colour charge, L is left and Y is the weak hypercharge. It is also Lorentz invariant, as required by special relativity. The field content of the SM is shown in Table 2.1. Notice that the gauge fields are not included in this table, more details on them are described in the following sections.

Field	$SU(3)_c$	$SU(2)_L$	$U(1)_Y$
Q_L	**3**	**2**	$\frac{1}{6}$
u_R	**3**	**1**	$\frac{2}{3}$
d_R	**3**	**1**	$-\frac{1}{3}$
L_L	**1**	**2**	$-\frac{1}{2}$
e_R	**1**	**1**	-1
H	**1**	**2**	$\frac{1}{2}$

Table 2.1: Field content of the SM.

2.1.1 Quantum Chromodynamics

Quantum Chromodynamics (QCD) is the theory of strong interaction, first completely written down by Fritzsch, Gell-Mann, and Leutwyler [7] and later on polished by Gross and Wilczek [8], and Politzer [9]. It is a non-abelian theory with $SU(3)_c$ symmetry, non-abelian simply means the generators within this theory are non-commutative. Quarks have three (N_c) different colour charges as it is part of the fundamental representation of $SU(3)$. The gauge boson for QCD is the gluon, it is in the adjoint representation which means that it has eight ($N_c^2 - 1$) colours.

There are two QCD parts in the Langrangian from Eq. 2.1.1: the QCD gauge field tensor and the interactions between quarks and gluons inside the covariant derivative D. The gauge field tensor part can be written as:

$$G^a_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu + g_s f^{abc} G^b_\mu G^c_\nu \ , \tag{2.1.2}$$

where G^a is the gluon field, g_s is the strong coupling constant and f^{abc} is the antisymmetric structure constant of $SU(3)$. The covariant derivative connecting quarks and gluons can be written as :

$$(D_\mu)_{ij} = \partial_\mu \delta_{ij} - i g_s G^a_\mu t^a_{ij} \ , \tag{2.1.3}$$

where t^a are the generators of $SU(3)$. Some important relations from these generators are:

$$\begin{aligned}
[t^a, t^b] &= i f_{abc} t^c \ , \\
t^A_{ab} t^A_{bc} &= C_F \delta_{ac} \ , \\
C_F &\equiv \frac{(N_c^2 - 1)}{2N_c} = \frac{4}{3} \ , \\
f_{ACD} f_{BCD} &= C_A \delta_{AB} \ , \\
C_A &\equiv N_c = 3 \ , \\
t^A_{ab} t^B_{ab} &= T_R \delta_{AB} \ ,
\end{aligned} \tag{2.1.4}$$

where C_F s the colour factor ("Casimir") associated with gluon emission from a quark, C_A is the colour factor associated with gluon emission from a gluon and $T_R = 1/2$ is the colour factor for a gluon to split into a $q\bar{q}$ pair. It is important to note that QCD governs quarks into bound states called hadrons at low energy through a property known as colour confinement and it is also the main source of radiation at high energies such as a collider collision.

2.1.2 Electroweak sector and the Higgs mechanism

The electroweak sector is based on the model first introduced by Glashow [10], Weinberg [11] and Salam [12]. This model describes electroweak interactions between gauge fields and matter fields for the gauge group $SU(2)_L \times U(1)_Y$. In the SM, the left and right-handed fermion have different transformation properties under different gauge groups. The subscript L for $SU(2)_L$ is included to emphasise the left chirality. We can see this from Table 2.1 where left-handed fermions are doublets (**2**) within $SU(2)$ and the right-handed fermions are all singlets (**1**). Note that the $U(1)$ hypercharges are also different between left and right handed fermions which means that parity is violated. Further discussion on the global symmetries of the SM is included later in this chapter.

As fermions with different chiralities transform differently, the terms in the Lagrangian associated with them are naturally different. For left-handed fermions, the covariant derivative includes interactions with both $SU(2)$ and $U(1)$ gauge fields. It can be written as:

$$D_\mu = \partial_\mu - igA^a_\mu \frac{\sigma^a}{2} - ig' Y_L B_\mu \ , \tag{2.1.5}$$

where σ^a are the Pauli matrices, and A^a_μ and B_μ are the gauge fields of $SU(2)_L$ and $U(1)_Y$ respectively. For the right handed fermions, only the weak hypercharge field is at play. The covariant derivative is then:

$$D_\mu = \partial_\mu - ig' Y_R B_\mu \ , \tag{2.1.6}$$

where $Y_{L,R}$ have intentionally been included as different variables to further emphasise that left- and right handed- fields have different hypercharges.

The chirality dependence on the field when interacting with the gauge groups also means that standard Dirac mass terms are forbidden. In addition, mass terms for the gauge boson fields ($M^2 V_\mu V^\mu$) are not gauge invariant. However, we know that the $W^\pm$ and Z gauge bosons are definitely not massless. The missing piece to include these masses is known as the BEH mechanism [13–15]. The inclusion of a complex scalar field with the famous "Mexican hat" shaped potential allows the acquirement of the vacuum expectation value (VEV). This spontaneously breaks the symmetry of the SM Lagrangian down to $SU(3)_c \times U(1)_{\text{EM}}$, also known as the electroweak spontaneous symmetry breaking (EWSB). The potential mentioned can be written as:

$$V(H) = -\mu^2(H^\dagger H) + \lambda(H^\dagger H)^2 \ , \tag{2.1.7}$$

where it will take the hat shape for $\mu^2, \lambda > 0$. When this condition is met, the minimum for this potential is:

$$|H| = v \text{ where } v = \sqrt{\frac{\mu^2}{2\lambda}} \tag{2.1.8}$$

This non-zero VEV breaks the $SU(2)_L \times U(1)_Y$ electroweak symmetry down to the $U(1)_{\text{EM}}$ conservation of electric charge symmetry. Subsequently, the Higgs doublet can be written in unitary gauge as:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + h \end{pmatrix} \ , \tag{2.1.9}$$

where h is the Higgs boson field. We can now retrieve mass terms for the gauge bosons and fermions. For the case of the gauge bosons, the squared covariant derivative within the broken Higgs phase can be expanded and we can define three massive vector bosons ($W^{\pm}, Z$) and one massless vector field orthogonal to Z (photon A).

$$\begin{aligned} W^{\pm}_{\mu} &= \frac{1}{\sqrt{2}}(A^1_{\mu} \mp A^2_{\mu}) \text{ with mass } m_W = g\frac{v}{2} \ , \\ Z^0_{\mu} &= \frac{1}{\sqrt{g^2 + g'^2}}(g' A^3_{\mu} - g B_{\mu}) \text{ with mass } m_Z = \sqrt{g^2 + g'^2}\frac{v}{2} \ , \\ A_{\mu} &= \frac{1}{\sqrt{g^2 + g'^2}}(g' A^3_{\mu} + g B_{\mu}) \text{ with mass } m_A = 0 \ . \end{aligned} \tag{2.1.10}$$

This aligns with observation from experiments where we have massive W and Z bosons while the photon remains massless.

One can now consider the covariant derivate from Eq. 2.1.5 in terms of the mass eigenstates shown above, it is written as:

$$\begin{aligned} D_{\mu} = \partial_{\mu} &- i\frac{g}{\sqrt{2}}(W^+_{\mu}\sigma^+ + W^-_{\mu}\sigma^-) - i\frac{1}{\sqrt{g^2 + g'^2}}Z_{\mu}\left(g^2\frac{\sigma^3}{2} - g'^2 Y\right) \\ &- i\frac{gg'}{\sqrt{g^2 + g'^2}}A_{\mu}\left(\frac{\sigma^3}{2} + Y\right) \ , \end{aligned} \tag{2.1.11}$$

where $\sigma^{\pm} = \frac{1}{2}(\sigma^1 \pm i\sigma^2)$.

The term associated with the photon shows that the coefficient for electromagnetic interaction as the electro e can be written as:

$$e = \frac{gg'}{\sqrt{g^2 + g'^2}} \ , \tag{2.1.12}$$

and that the electric charge quantum number Q is:

$$Q = \frac{\sigma^3}{2} + Y \ . \tag{2.1.13}$$

In addition to the electromagnetic interaction, we can define the weak mixing angle θ_w, also known as the Weinburg angle [11]. It is the angle that appears when transforming between the (A^3, B) and (Z, A) bases:

$$\begin{pmatrix} Z \\ A \end{pmatrix} = \begin{pmatrix} \cos\theta_w & -\sin\theta_w \\ \sin\theta_w & \cos\theta_w \end{pmatrix} \begin{pmatrix} A^3 \\ B \end{pmatrix} \ , \tag{2.1.14}$$

the individual terms can be written as:

$$\cos\theta_w = \frac{g}{\sqrt{g^2 + g'^2}} \ , \ \sin\theta_w = \frac{g'}{\sqrt{g^2 + g'^2}} \ . \tag{2.1.15}$$

With the Weinburg angle, Eq. 2.1.5 can be rewritten as:

$$D_\mu = \partial_\mu - i\frac{g}{\sqrt{2}}(W_\mu^+\sigma^+ + W_\mu^-\sigma^-) - i\frac{g}{\cos\theta_w}Z_\mu\left(\frac{\sigma^3}{2} - \sin^2\theta_w Q\right) - ieA_\mu Q \ , \tag{2.1.16}$$

where $g = \frac{e}{\sin\theta_w}$. This form of the covariant derivative implies the couplings of all weak bosons can be described with two parameters: the well-measured electron charge e and the new mixing angle θ_w. The coupling induced by W and Z will also involve the masses of these bosons. However, the masses are not independent where $m_W = m_Z \cos\theta_w$. Subsequently, all effects of W and Z exchange on tree level can be written in terms of three basic parameters: e, θ_w and m_W. Notice that fermion mass terms have not been mentioned although they are crucial for the next section.

2.2 Flavour

The quarks and leptons of the SM are organized into three generations. The fundamental properties between the generations are practically identical apart from their masses. In these three generations, six different types of quarks and leptons are also addressed as different flavours. The assortment of flavours give rise to a mix of phenomenological interactions. Heavy flavour physics refers to studies of the heaviest quarks, that is the quarks in the third generation (top and bottom), and there is also some interest in the second generation (charm and strange). This is the main focus of this thesis.

Particles from one generation can alter their flavour through two types of processes: the first is a tree-level process interacting with a $W^\pm$ boson, the other type is called a flavour changing neutral current (FCNC) interactions. The second type is forbidden at tree level within the SM; loop suppression applies such that they are much rarer compared to the first type.

2.2.1 CKM matrix

In the previous section, we mentioned that quarks acquire masses through the BEH mechanism. The Lagrangian terms representing this phenomenon are known as Yukawa Interactions. They are interactions between the quark and the Higgs field. They can be written as:

$$\begin{aligned} \mathcal{L} &\supset Y^u_{ij}\overline{Q^i_L}\tilde{H}u^j_R + Y^d_{ij}\overline{Q^i_L}Hd^j_R + \text{h.c.} \\ &\supset \frac{v}{\sqrt{2}}Y^u_{ij}\overline{u^i_L}u^j_R + \frac{v}{\sqrt{2}}Y^d_{ij}\overline{d^i_L}d^j_R + \text{h.c.}\ , \end{aligned} \tag{2.2.1}$$

where $\tilde{H} = i\sigma^2 H^*$ and i, j are the generation indices. Note that the Higgs field written in unitary gauge from Eq. 2.1.9 is applied and terms without the VEV are dropped. We can find the quark masses by applying singular value decomposition into the mass basis of Y. This is needed because Y are not guaranteed to be diagonal. We find:

$$M^u = \frac{v}{\sqrt{2}}U^u_L Y^u (U^u_R)^\dagger \text{ and } M^d = \frac{v}{\sqrt{2}}U^d_L Y^d (U^d_R)^\dagger\ , \tag{2.2.2}$$

where $U^{u,d}_{L,R}$ are unitary matrices. The quark mass matrices $M^{u,d}$ are now diagonalised such that $M^u = \text{diag}(m_u, m_c, m_t)$ and $M^d = \text{diag}(m_d, m_s, m_b)$. There are two important bases for Y: one where the masses are diagonal, called the mass basis (Eq. 2.2.2), and the other where the $W^\pm$ interactions are diagonal, called the interaction basis. The $W^\pm$ basis arose from the fact that W bosons couple up and down type quarks together. Note that these two bases do not produce the same result in flavour-changing interactions. The change of basis can be written as:

$$\overline{u_L}\gamma^\mu d_R W^+_\mu \rightarrow \overline{u_L}(U^u_L)\gamma^\mu(U^d_L)^\dagger d_R W^+_\mu \equiv \overline{u^i_L}\gamma^\mu V_{ij} d^j_R W^+_\mu \tag{2.2.3}$$

where $V_{ij} \equiv U^u_L(U^d_L)^\dagger$ is the Cabibbo–Kobayashi–Maskawa (CKM) matrix [16,17]. In other words, the CKM matrix is the rotation between these two bases. As the CKM matrix connects the flavours as shown in Eq. 2.2.3, a common way of representing it is as follows:

$$V = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \approx \begin{pmatrix} 0.97 & 0.23 & 0.0037e^{-1.1i} \\ -0.22 & 0.97 & 0.042 \\ 0.0086e^{-0.39i} & -0.041 & 1 \end{pmatrix} \tag{2.2.4}$$

Each of the matrix element in the CKM can only be determined through experimental measurements. However, additional improvements can be made using global fits which combine theory calculations and experimental data. The values shown in Eq. 2.2.4 are approximations of fit results from CKMFitter [18, 19]. There are two common ways in parametrising the CKM, the standard way through three mixing angles and one phase [20] or the Wolfenstein way [21]. The standard parametrisation scheme arose from the fact that a general 3 × 3 unitary matrix has nine degrees of freedom and they can be separated into six phases and three real parameters. Additionally, all but one of the phases can be absorbed into the quark fields leaving three real parameters $(\theta_{12}, \theta_{13}, \theta_{23})$ and just one phase (δ_{13}). The real parameters are also known as mixing angles between different generation of quarks. The CKM under standard parametrisation is written as:

$$V = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{i\delta_{13}} \\ -s_{13}c_{23} - c_{12}s_{23}s_{13}e^{i\delta_{13}} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta_{13}} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta_{13}} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta_{13}} & c_{23}c_{13} \end{pmatrix} , \tag{2.2.5}$$

where $c_{ij} = \cos\theta_{ij}$ and $s_{ij} = \sin\theta_{ij}$. The Wolfenstein parametrisation originated from an attempt to represent the CKM through a small parameter expansion $V_{us} \approx 0.2$ up to order $\mathcal{O}(\lambda^4)$. The CKM under this parametrisation is written with four parameters (λ, A, ρ, η), and has the form:

$$V = \begin{pmatrix} 1 - \lambda^2/2 & \lambda & A\lambda(\rho - i\eta) \\ -\lambda & 1 - \lambda^2/2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4) \ . \tag{2.2.6}$$

This alternative parametrisation scheme displays important features of the CKM as shown in the numerical version in Eq. 2.2.4. It is almost an identity matrix and therefore flavour transitions are suppressed. Such transitions between the first and second generations are less suppressed compared to the first and third generation transitions. Also, the complex elements (V_{ub}, V_{td}) are both $\mathcal{O}(\lambda^3)$ which means they are heavily suppressed.

Chapter 3

Machine Learning in Particle Physics

Machine learning (ML) is one of the fastest growing fields since the late 20th century. Data driven solutions spawn across most if not all scientific studies. The idea behind ML is relatively simple; can a computer perform a task without being explicitly programmed to do so. There are generally four types of ML, supervised, unsupervised, semi-supervised and reinforcement learning. For supervised and unsupervised learning, they differ by whether the algorithm is given labels of the training data as a reference or not. Semi-supervised learning is a combination between supervised and unsupervised learning where a small amount of data contain labels but the majority are not labelled. Reinforcement learning is the most unique out of the four types where it can be thought of as an agent trained to make a sequence of decisions within a predefined environment. Independent of these four types, ML is used to solve two general problems, regression and classification. The goal of a regression problem is to statistically fit a model such that the real and continuous target distribution can be predicted given further input data. On the other hand, a classification problem has a categorical target e.g. a colour or types of flower, and the goal is then to draw a conclusion on whether a set of input value represents a certain category. This chapter focuses on introducing key concepts of machine learning and their applications in particle physics. In particular, we will concentrate on decision trees and neural networks as they are the methods used throughout this thesis. Ensembling will also be discussed as a method to improve performance for both algorithms.

Machine learning applications in particle physics have evolved rapidly in the past decades. The most common application has been event classification type problems where the algorithm decides if a particle collision event belongs to a certain physical process, the most refined usage is certainly quark gluon classification where a vast array of methods have been developed to tackle this traditionally difficult problem due to their similarity in fundamental properties [22–24] . Even though there were more interest in classification problems early on, popularity in regression has caught on as simplification for Monte Carlo simulations became more effective [25, 26]. A complete review of ML applications can be found in Ref. [27–30]. This chapter is inspired by Ref. [31].

3.1 Model and parameters

Before we introduce any ML algorithms, some simple concepts on model fitting are reviewed here. For the purpose of this thesis, we shall focus on supervised binary classification problems, we have:

$$f(x) : \mathbb{R}^d \to \mathbb{R}, x \in \mathbb{R} \mapsto y^* \in [0,1] \ , \tag{3.1.1}$$

where f is some ML algorithm which takes in a d-dimensional vector x and outputs a real number y^* between 0 and 1. Note that $f(x)$ is used interchangeably with model and x is often referred to as the input variables or features throughout this thesis. The algorithm f learns to become a classifier from the training dataset D which contains n pairs of vectors x and labelled output y such that:

$$D = \{(x_i, y_i) | x \in \mathbb{R}^d, y \in [0,1]\}, |D| = n \ . \tag{3.1.2}$$

In practice, f divides the d-dimensional input space into two regions for the two classes. The task of finding this partition can be generalised as training the model to find the best set of parameters $\{\theta\}$ that fits the data x_i to label y_i. In order to train any model, an objective function is required and it can be written as:

$$\text{obj}(\theta) = L(\theta) + \Omega(\theta) \ , \tag{3.1.3}$$

where L is the loss term and Ω is the regularisation term. The choice of function for the loss term varies depending on the task. A typical loss function for binary classification is the cross-entropy. The regularisation term is crucial in controlling the complexity of the algorithm. The process of training a model is then to minimise the objective function.

There are parameters optimised for the model as shown in Eq. 3.1.3 and there are also hyperparameters defining the structure/complexity of the algorithm. The training data provided to the algorithm optimises the performance based on the hyperparameters but a problem could occur where the model is fitted exactly to the training data and therefore fails to generalise to unseen data, such phenomenon is known as overfitting. The regularisation term has an effect in avoiding overfitting.

A data set often contains three parts: training, validation and testing sets. The purpose of having a validation set is to help visualise whether overfitting has occurred. This is common where the values of the objective function are computed for both the training and validation sets, a divergence between the values indicates overfitting has occurred. In order to evaluate the performance, the testing set is typically generated independently from the training and validation sets such that the algorithm could not have learnt from these data during the learning and optimisation phases. Once a model is well optimised, it can be used for inference which simply means the utilisation of this model for making predictions on real data leading to actionable outputs.

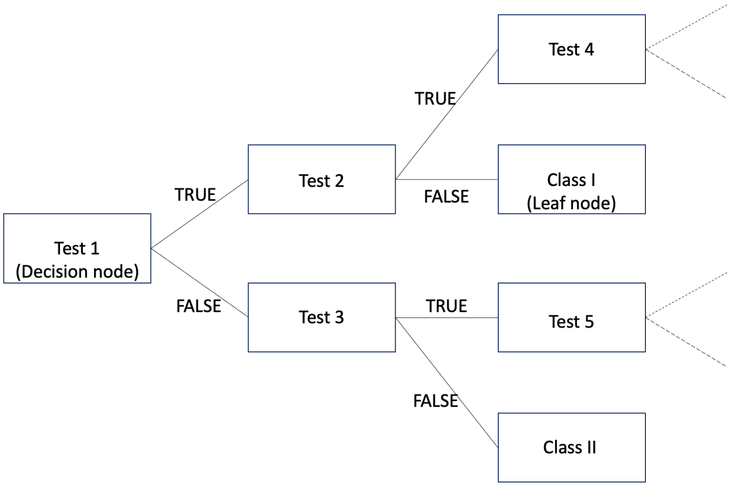

Figure 3.1: Schematic diagram of a decision tree.

3.2 Boosted decision tree

Boosted decision tree (BDT) [32] is a staple piece in the machine learning arsenal for various fields as it is effective yet simple and clearly interpretable. It is important to start with the basic unit, what is a decision tree?

3.2.1 Decision tree

A decision tree [33] is a device that assigns a class number/decision based on the input variables. The tree is built with decision nodes and leaf nodes. The decision nodes are tests on various parts of the input vector. The tests are iteratively constructed and updated throughout the learning process, they can also be thought of as making cuts on the input features. The leaf nodes are where the class number is decided. A path leading to a particular result is known as a branch, this can be understood as a conditional probability for the result to be a certain class. A diagram of what a tree might look like is given in Fig. 3.1.

The rules in building and pruning a tree slightly differ between different tree algorithms, we follow the procedures from the classification and regression trees (CART) [34] as they are the tree models used throughout this thesis. The rules are as follows:

- The decision node creates tests based on the input variables such that the best partition between the two classes is obtained.

- Each decision node recursively splits into child nodes and the first item is applied.

- The splitting stops when no further information/performance gains are detected or some pre-defined ending conditions are met.

The method used by CART to detect information gains is known as Gini impurity [35]. It is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the training dataset. The Gini impurity for a dataset or feature can be written as:

$$I_G(p) = \sum_{i=1}^{J} p_i(1-p_i) \ , \tag{3.2.1}$$

where J is the number of classes and p_i is the fraction of data points in class i within the dataset. A score can then be calculated for each branch of the tree through this measure and a weighted sum of the whole tree determines if additional information has been gained. This process is analogous to the entropy method shown in Eq. 3.2.2 where the weighted sum of score is replaced with the entropy E. A tree on its own is a weak learner, often naive and prone to problems such as overfitting. A possible solution for weak learners such as trees to avoid this problem is to have an ensemble of trees. A comprehensive review of tree methods can be found in Ref. [36].

$$E = \sum_{i=1}^{J} -p_i \log_2 p_i \ . \tag{3.2.2}$$

3.2.2 Boosting

Using an ensemble of trees is an effective scheme to reduce variance and bias on the predictions and therefore obtain a relatively stable predictor. There are two main streams of techniques in creating ensembles, bagging and boosting. This thesis has

a stronger focus on boosting ensembles as they are more commonly used in particle physics. Bagging refers to training multiple models with randomly chosen subsets of the original training dataset and the combined predictions would have lower variance compared to a single tree model. One obvious advantage of bagging is that the models are independent of each other in terms of training, allowing for parallel computing. A famous bagging ensemble tree method is the Random Forest [37].
On the contrary, boosting trains the models sequentially which results in one strong learner. Within the sequence of trees, early models are simple and any misclassified samples of the input data would be highlighted with an increased weight. Subsequently, the later models would have a stronger focus on those weighted samples leading to better results overall. There have been multiple techniques developed to optimise the boosting process, the most common one is called gradient boosting [38, 39] where a gradient descent algorithm can optimise a given differentiable loss function to guide the construction of future trees. One well known package for gradient boosting tree methods is called `XGBoost` [40]. Note that ensemble techniques are applicable to algorithms other than trees for the same benefits.

3.3 Deep learning

Deep learning (DL) refers to neural network (NN) related ML algorithms throughout this thesis. Networks of non-linear elements, interconnected through adjustable weights, play a prominent role in machine learning. These type of object is known as neural networks. They got the name because of the vague resemblance to networks of biological neurons. There are many methods in how a network can be connected and the type of basic units used to construct it. In this thesis, we will focus on the fully connected neural networks, which we will call NN, and an extended version called Bayesian neural networks (BNN) [41, 42].

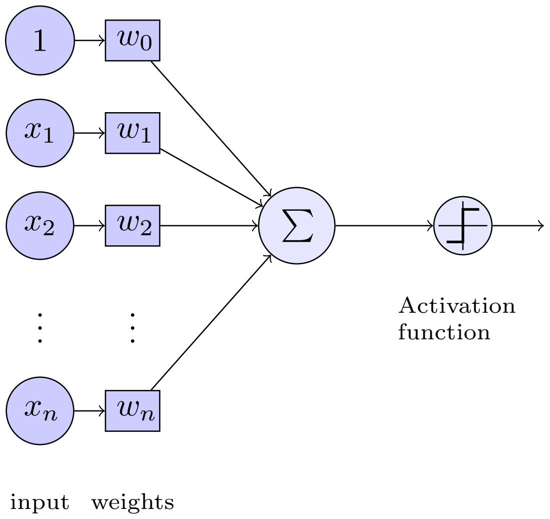

Figure 3.2: Schematic configuration of a perceptron.

3.3.1 Perceptron

A perceptron is the fundamental building block of a neural network, it was first popularised by Minsky and Papert in 1969 as a type of threshold logic unit [43, 44]. It outputs a decision given an input data point $x = (x_1, x_2, ..., x_d)$, the calculation involved can be written as:

$$\text{output}(x) = \text{act}(\sum_{i=0}^{d} x_i \cdot w_i + \text{bias}) \ , \tag{3.3.1}$$

where w_i are the weights which controls the relative importance of each feature, the bias term is typically referred to as the zero-th data point where bias $= 1 \cdot w_0$ and act is short for the activation function. A diagrammatic version of this calculation is shown in Fig. 3.2. A vast collection of activation functions have been developed over the years, we will focus on the sigmoid function here as it is the most common choice for the output of a binary classifier. We can see why it is a good choice from Fig. 3.3, it is an “S” shaped bounded and differentiable curve which returns a real number between 0 and 1. The bounded nature acts conveniently as a probability like measure. We have the basic structure of a perceptron defined, the learning process is then to find the best weights, as mentioned in Section 3.1, such that the best partition between the two classes can be obtained. Recall from Eq. 3.1.3, any model requires an objective function and the binary classifiers typically use binary

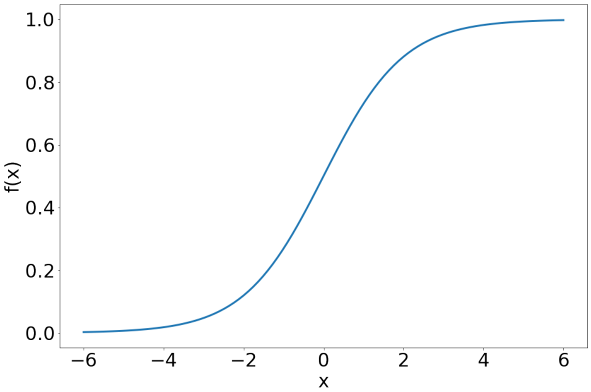

Figure 3.3: Sigmoid function.

cross-entropy as the loss function. It can be written as:

$$L(x, y, w) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i(x, w)) + (1 - y_i) \cdot \log(1 - \hat{y}_i(x, w)) \ , \tag{3.3.2}$$

where N is the number of data points, y_i is the i-th label and $\hat{y}_i$ is the i-th prediction. The log terms emphasis small difference between the prediction and the label. We know that $\hat{y}_i$ depends on the weights w_i from Eq. 3.3.1, the learning process is then to find the optimal weights such that L is minimised.

Gradient descent algorithms are part of the family of automated differentiation libraries. They are the standard methods for optimising weights within machine learning. The principle behind gradient descent is simply the chain rule of differentiation. The weights are updated like:

$$w(t + 1) = w(t) - \eta \nabla_w L \ , \tag{3.3.3}$$

where the weights, $w = (w_1, w_2, ..., w_j)$, update at timestep $t + 1$ depending on the weights from the current timestep/iteration t and the gradient of the loss function w.r.t. weights multiplied by the learning rate η. The fact that the gradient of the loss function is involved highlights the importance of the differentiable property. The learning rate is one of the many optimisable hyperparameters for a ML model, it is important because it essentially controls the step size of the gradient decent. The goal of the descent is to find the global minimum, a learning rate too large could cause the model to converge into a suboptimal minimum and a learning rate too small could stop the model from leaving a suboptimal local minimum. The common practice is to set an initial learning rate and allow it to evolve with future iterations depending on the gradient descent algorithm. The most popular gradient descent algorithm in recent years is called adaptive moment estimation (`Adam`) [45], it changes the learning rate based on exponentially decaying averages of past squared and non-squared gradients along with the concept of momentum [46] which helps guide the direction of the decent. An overview of various gradient descent algorithms can be found in Ref. [47].

3.3.2 Including regularisation

The previous section described the learning process of a perceptron and how the weights can be updated to minimise the loss function. However, we know from Eq. 3.1.3 that there is more than just minimising the loss function. The regularisation term plays an important role in stabilising the model in order to avoid overfitting. There are various methods in applying regularisation, we will focus on L1 and L2 regularisation. There are other algorithm specific methods which will be mentioned in the corresponding sections. L1 and L2 regularisation, also known as LASSO and ridge regression respectively, can be thought of as a penalty term added onto the objective function. L1 adds on the sum of the absolute value of the weights and L2 adds on the sum of the squared magnitude of the weights. They are written as:

$$\mathrm{L1} = \lambda \sum_{j=0}^{m} |w_j| \ , \ \mathrm{L2} = \lambda \sum_{j=0}^{m} w_j^2 \ , \tag{3.3.4}$$

where λ is a hyperparameter functioning like the strength of the regularisation. These penalty terms drive the weight of less important features down such that the model focuses on the useful features. Note that L1 can reduce the weight to zero while L2 can only take the weight close to zero.

The learning process with the regularisation term included is akin to the description

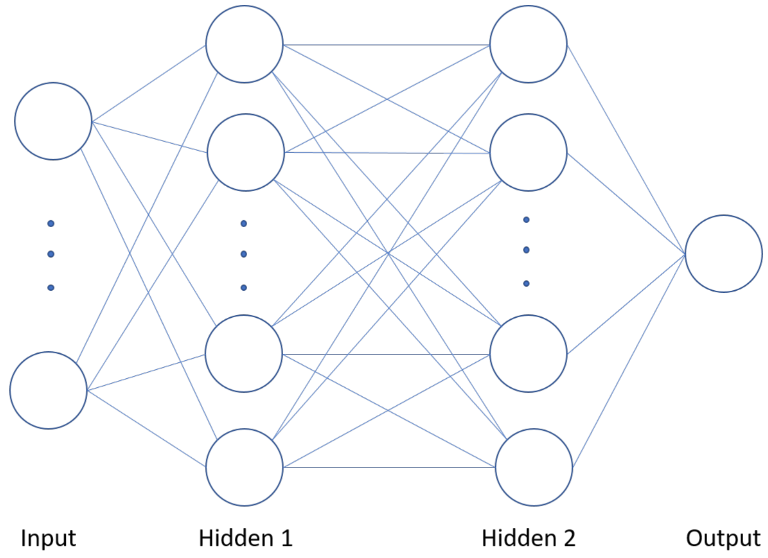

Figure 3.4: A fully connected neural network with 2 hidden layers.

in Eq. 3.3.3. Using L2 as the example, the updated weight has the form:

$$w(t+1) = w(t) - \eta \nabla_w L - 2\eta\lambda w(t) \ . \tag{3.3.5}$$

3.3.3 Fully connected neural network

The classification power of a single perceptron is limited as the task become more complicated. The next intuitive step is to connect multiple perceptrons together in parallel, the resultant object is known as a dense layer. A neural network (NN) can then be constructed by connecting layers together in series. The structure of any NN typically consists of at least three layers, the input layer, output layer and hidden layer(s) sandwiched between them as shown in Fig. 3.4. A deep neural network is simply a network with more than one hidden layers. It is a fully connected neural network when all neurons/perceptrons from two neighbouring layers are connected. NNs have established a reputation for approximating highly non-linear functions well, it is known that a network with two hidden layers where their activation functions are non-linear can approximate any continuous function of n real variables given an infinite number of neurons and training data [48]. The non-linearity comes from the activation function as the action would otherwise just be a series of dot products as shown in Eq. 3.3.1. We have already encountered the sigmoid function above, other popular activation functions include ReLU [49] and the hyperbolic tangent (tanh). This non-linearity increases with the complexity of the network, this can be seen from the output of the overall network given as:

$$\text{output}_{\text{NN}}(x) = f^o(f^h(f^{h-1}(...f^1(x))) \ , \tag{3.3.6}$$

where each f represents an output calculated according to Eq. 3.3.1, $f^o(...)$ is the output of the output layer and h is the total number of hidden layers. The choice of activation function for each layer is highly problem dependent, we mentioned the sigmoid function being common for the output of binary classification but it would be an unreasonable choice for a regression problem which typically uses a linear output activation function. There are also hyperparameters associated with the architecture of the network for example the number of hidden layers or the number of neurons per layer. The learning process for a neural network is basically an extended version of the gradient descent described in the previous section. The weight optimisation process is known as back-propagation where the chain rule of differentiation is used to calculate the Jacobian ∇_W with respect to each neuron in the network. The vector of weights from Eq. 3.3.3 becomes a matrix updating under the same principles. This process can be written as:

$$W(t+1) = W(t) - \eta \nabla_W L \ , \tag{3.3.7}$$

where W is a matrix of weights where its dimension depends on the complexity of the network.

In terms of regularisation, the L1 and L2 described in Section 3.3.2 are fully compatible with NNs. Other common techniques include the dropout layer [50] and early stopping. Each dropout layer has a rate which is a tunable hyperparameter. During training, some number of layer outputs are randomly ignored or “dropped out” based on the rate. This has the effect of making the layer look like and be treated like a layer with different number of nodes and connectivity to the prior layer. In effect, each update to a layer during training is performed with a different “view” of the configured layer. In addition, the training process will seem much noisier, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs. This conceptualisation suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model training process more robust.

On the other hand, early stopping is much more straightforward, it is a mechanism monitoring the training and validation loss. The training process is stopped when the two losses deviates by a certain level chosen based on the target training precision. This ensures the model is not overfitted and that the best model weights are obtained for inference.

3.3.4 Bayesian neural network

Development in different neural network architectures is arguably the fastest growing field of research right now in ML. Different networks have been created for different purposes such as the computer vision related convolutional networks [51] and the natural language processing focused recurrent networks [52]. Bayesian neural network (BNN) is an extension of the fully connected neural network, it can be thought of as a neural network that trains based on the Bayes’ theorem.

Assume a neural network is viewed as a probabilistic model $p(y|x, w)$ where y are the labels, x are the inputs and w are the weights, training the model with dataset D is the same as constructing the likelihood function $p(D|w)$ and maximising it based on w, this process is also known as maximum likelihood estimation (MLE). This likelihood function corresponds to the cross-entropy shown in Eq. 3.3.2 and the maximisation process is the same as optimising the loss function.

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \tag{3.3.8}$$

From Bayes’ theorem, shown in Eq. 3.3.8, we know that the likelihood function $p(D|w)$ multiplied by a prior distribution $p(w)$ is proportional to $p(w|D)$. Therefore, if we maximise $p(w|D)$, this gives the maximum a posteriori (MAP) estimate of w. The advantage of using MAP instead of MLE is that the model is naturally regularised and hence is equivalent to optimising the loss function with the regularisation term added on. The type of regularisation depends on the choice of distribution chosen as the prior. Note that MLE and MAP both give point estimates of the weights and biases. If the full posterior distribution over the parameters is available, one can make predictions while accounting for the uncertainty on the weights. This can be achieved from computing the predictive posterior distribution:

$$p(y|x, D) = \int p(y|x, w)p(w|D)dw \ . \tag{3.3.9}$$

Note that this computation is the same as averaging predictions from an ensemble of standard neural networks. The method up till this point sounds ideal, however there are no analytical solutions for the true posterior $p(w|D)$. One way around this is to approximate the true posterior with a surrogate variational distribution $q(w|\theta)$ where $\theta = (\mu, \sigma)$, the mean and standard deviation of the variational posterior. The learning for this distribution is done through minimising the Kullback-Leibler divergence (KL) given in Eq. 3.3.10, which measures the similarity between two probability distributions, between the surrogate $q(w|\theta)$ and the true posterior $p(w|D)$ as a function of the prior and the likelihood function through Bayes' theorem.

$$\mathrm{KL}[q(w|\theta), p(w|D)] = \int dw \ q(w|\theta) \log \frac{q(w|\theta)}{p(w|D)} \ . \tag{3.3.10}$$

We have shown a mathematical overview of the BNN but how does it apply in the context of ML? Recall an object of multiple perceptron-like neurons connected in a parallel fashion is known as a dense layer. One of the two key differences between BNN and NN is the type of layer used to construct the network. BNN commonly utilises dense flipout layers instead of the standard dense layers. Flipout layers perform a Monte Carlo approximation of the posterior distribution integrated over the weights and biases [53]. In other words, the weights and bias are no longer points but distributions characterised by mean μ and standard deviation σ as described in $q(w|\theta)$. There are a variety of prior distributions available but we will focus on the Laplace and the Gaussian distributions. These two distributions are mentioned because model trained with these priors are equivalent to models with L1 and L2 regularisations applied [54]. The learning process needs to include a KL-divergence term which is also the second major difference. A complete derivation of this objective function can be found in Ref. [55].

The weights updating process for BNNs is similar to a fully connected NN where back-propagation is applied to compute the gradient of the mean μ and standard deviation σ. It is important to note that a re-parametrisation trick is needed for this back-propagation to work. This is because the gradient computed naturally would otherwise be very close to zero. The trick is to sample from a parameter-free distribution and then transform the sampled ϵ with a deterministic function $t(\mu, \sigma, \epsilon)$ for which a gradient can be defined.

Once the model is ready for inference, each prediction for the same input will be different as we would be sampling from a distribution of weights instead of the point values of weights. This allows the user to make n predictions for each data point in the test set and in turns estimate the epistemic uncertainty arising from the neural network weights. This process is also known as Bayesian inference.

3.4 Metrics

This section reviews some common metrics used when evaluating performances of the ML algorithms in particle physics. For a binary classifier taking as input the multidimensional features of an event, and returning a classifier output which is a single number, $\zeta \in [0,\, 1]$. Events with classifier output $\zeta \sim 1$ are likely to be signal while events with $\zeta \sim 0$ are likely to be background. We define our signal (fiducial) region through a cut on the classifier output. All events with $\zeta > \zeta_{\mathrm{cut}}$ are classified as signal events. Events which are correctly classified as signal events are denoted true positive (TP) events, while background events which are incorrectly classified as signal events are denoted false positive (FP) events. Note that this description is general to any binary classifier.

Standard performance metrics in ML are the receiver operating characteristic (ROC) curve, i.e. the true positive rate (TPR, signal acceptance) as a function of the false positive rate (FPR, background acceptance), and the corresponding area-under-curve (AUC), the integral of the ROC curve. TPR and FPR are given as:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \ , \ \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \ . \tag{3.4.1}$$

It is also customary to plot the inverse of the FPR as a function of the TPR. The optimal classification threshold (ζ_{cut}) is often selected by the best statistical significance. We now discuss this in more detail.

The best estimate for the number of signal events can be written as:

$$S = N - B \ , \tag{3.4.2}$$

where N is the total number of events, B is the number of background events and S is the number of signal events. The uncertainty for this estimate is:

$$\sigma^2(S) = \sigma^2(N) + \sigma^2(B) = N + \sigma^2(B) \tag{3.4.3}$$

where $\sigma(S)$ is the standard deviation of S and N is characterised by a Poissonian fluctuation such that $\sigma^2(N) = N$. In addition, the sample size is assumed to be large for B such that $\sigma(B)$ is small and negligible. The significance is then:

$$\frac{S}{\sigma(S)} = \frac{S}{\sqrt{N}} = \frac{S}{\sqrt{S+B}} = \frac{\mathrm{TP}}{\sqrt{\mathrm{TP}+\mathrm{FP}}} \tag{3.4.4}$$

where S is the equivalent to TP and B is the same as FP.

There are two ways forward to improve significance as a metric, the first is to calculate it based on expected events. This is a common technique for particle physics studies because the ML test result retains statistical accountability in an experimental environment. The conversion between standard ML TP to the expected number of true positive events $\mathrm{TP}^{\mathrm{exp.}}$ is as follow:

$$\mathrm{TP}^{\mathrm{exp.}} = \epsilon_{\mathrm{TP}}^{\mathrm{NN}} * \epsilon_{\mathrm{detector}} * \sigma_{\mathrm{signal}}^{\mathrm{MC}} * L \ , \tag{3.4.5}$$

where $\epsilon_{\mathrm{detector}}$ is the efficiency from applying detector level cuts, $\epsilon_{\mathrm{TP}}^{NN}$ is the signal efficiency TPR at the optimal ζ_{cut}, $\sigma_{\mathrm{signal}}^{MC}$ is the cross section of the signal process obtained from the MC data generation and L is the luminosity. The expected background $\mathrm{FP}^{\mathrm{exp.}}$ is calculated in the same fashion with the efficiency and cross section replaced for the background process. Note that if there is more than one background process,

$$\mathrm{FP}^{\mathrm{exp.}} = \sum_{i \neq \mathrm{signal}} \mathrm{FP}_i^{\mathrm{exp.}} \ . \tag{3.4.6}$$

The other method is to remove dependence on the data sample size from the significance, this new quantity is known as the significance improvement $\hat{\sigma}$. It is essentially the significance normalized to its value at the baseline selection.

$$\hat{\sigma} = \frac{\sigma}{\sigma_{\mathrm{baseline}}} \ , \tag{3.4.7}$$

where $\sigma_{\mathrm{baseline}}$ is given as the truth number of signals / truth number of signal and background. A significance improvement greater than one signals a performance increase compared to the baseline selection. Plotting the significance improvement as a functions of the true positive rate defines the significance improvement characteristic (SIC) curve [56].

Chapter 4

Machine-learning Approaches to Inclusive $|V_{ub}|$ Determinations

Studies on elements of the CKM matrix is an important part of flavour physics as they are central to testing the CKM picture of quark mixing and CP violation. This chapter focuses on the determination of the least known element of the CKM matrix, $|V_{ub}|$. We explore the usage of machine learning (ML) techniques as multivariate analyses, in particular the impact from the choice of Monte Carlo (MC) generator used to simulate collision events, the set of features employed in training the algorithms and the efficiency of such methods from not just a ML point of view but also a physics standpoint. This chapter is based on Ref. [3, 57].

4.1 Semi-leptonic B decay

Precise determination of $|V_{ub}|$ is crucial in testing the flavour structure of the SM. The measurement of $|V_{ub}|$ is particularly interesting since it is one of the smallest and least known element. Common extraction techniques involve studying semi-leptonic decays of B mesons.

The B meson[2], lightest particle containing a b quark, decays weakly. In general, the

[2] The flavour or charge of the B mesons are not specified unless specifically mentioned.

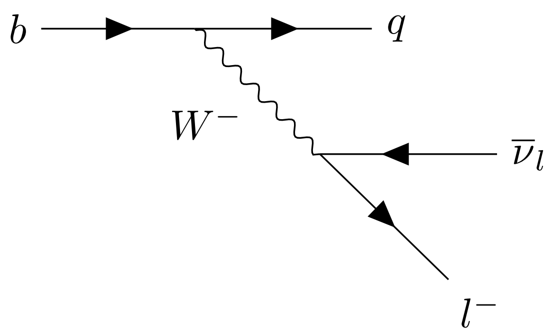

Figure 4.1: b quark decaying through charge current interaction into quark $q \in \{c, u\}$.

semi-leptonic decay B decay can be written as:

$$B \to X_q l\nu \ , \tag{4.1.1}$$

where the final states consist of a lepton-neutrino pair (l, ν) and hadronic system X_q, with q being either c or u quark. Note that X_q does not need to be a single hadron, it can also be a group of hadrons. From the perspective of partons, the b $(\bar{b})$ quark undergoes a flavour changing charge current interaction through the emission of a W^- (W^+) gauge boson and becomes a q $(\bar{q})$ quark like $b \to ql^-\bar{\nu}$ $(\bar{b} \to \bar{q}l^+\nu)$. This decay is shown in Fig. 4.1. The remaining valance quark in the B meson can be assumed to be a spectator to good approximation. In order to thoroughly describe the dynamics of the decay, the non-perturbative effects between the B meson and final state X_q should be accounted for. However, since there is no strong interaction between the lepton-neutrino pair and the hadronic final state X_q, it is possible to factorize the strong and weak interaction contributions and treat them separately. Consequently, such decays permits clean extraction of the CKM-matrix elements $|V_{qb}|$ and the study of non-perturbative effects.

The constituent of a B meson include a heavy b quark, $m_b \approx 4.5 - 5\text{GeV} \gg \Lambda_{\text{QCD}}$, and a light u or d quark with masses much smaller than Λ_{QCD}. This configuration indicates that the framework of Heavy Quark Effective Theory (HQET) can be applied to describe its dynamics. m_b with such a high value has two implications: perturbative QCD is valid in this description as the strong coupling constant is at the energy scale of m_b $\alpha_s(m_b) \approx 0.2$. Furthermore, $\Lambda = \Lambda_{\mathrm{QCD}}/m_b \approx 0.1$ is a reasonable expansion parameter for non-perturbative effects. Subsequently, a systematic expansion of QCD in powers of Λ and α_s can be performed within the HQET framework. The description above on semi-leptonic B decay applies when q is either c or u. The remainder of this chapter switches focus back to only $b \to u$. The effective SM Lagrangian for these decays can be written as:

$$\mathcal{L}_{\mathrm{eff}} = \frac{-4G_F}{\sqrt{2}} V_{ub} (\bar{u}\gamma_\mu P_L b)(\bar{\nu}\gamma^\nu P_L l) + \mathrm{h.c.}\ , \tag{4.1.2}$$

Notice that the W boson has been integrated out of the Lagrangian as this type of decays occur at a typical scale of m_b. The W propagator is largely driven by the $m_W = 80$ GeV and therefore the interaction term is replaced by the effective coupling $4G_F V_{ub}/\sqrt{2}$ and the four-fermion operator. G_F is the Fermi constant, V_{ub} is the CKM-matrix element and $P_L = (1 - \gamma_5)/2$ is the projection operator on the left-handed part of the spinors.

4.1.1 Inclusive and exclusive

There are two general paths in extracting $|V_{ub}|$ from semi-leptonic B decays, exclusive or inclusive. Exclusive measurements target a specific hadronic decay mode such as $\bar{B} \to \pi l \bar{\nu}$. They are typically experimentally cleaner in terms of signal-to-background ratios as the processes are carefully selected. However, the exclusive branching fraction is typically only a few percent of that for inclusive decays. Inclusive measurements on the other hand consider all possible hadronic final states related to the target flavour transition.

From the theoretical standpoint, the inclusive $B \to X_u l \nu$ decay rate would offer the cleanest extraction of $|V_{ub}|$. The computation of the inclusive differential decay rate involves integrating over all possible hadronic final states X_u. This integration is applied over three independent variables, the choices shown here are the energy of the lepton (E_l), the invariant mass of the hadronic final state (M_X) and the invariant mass of the lepton-neutrino system (q^2) with $q = p_l + p_\nu$. The triple differential decay rate under these three variables is given as:

$$\begin{aligned}\frac{d^3\Gamma}{dq^2dE_ldM_X}\frac{G_F^2|V_{ub}|^2}{16\pi^2}(m_B-P_+)((P_--P_l)(m_B-P_-+P_l-P_+)\mathcal{F}_1 \\ +(m_B-P_-)(P_-+P_+)\mathcal{F}_2 \\ +(P_--P_l)(P_l-P_+)\mathcal{F}_3)\ ,\end{aligned} \tag{4.1.3}$$

with $P_l = m_B - 2E_l$ and $P_\pm = E_X \mp |\mathbf{p}_X|$, where m_B is the B meson mass, E_l is the energy of the lepton, E_X and p_X are the energy and three-momentum of the hadronic system. The $\mathcal{F}_i$ are the structure functions of the B meson which include shape functions and corrections to the strong coupling constant.

The evaluation of this integral is described by a local Operator Product Expansion (OPE) in inverse powers of the b-quark mass [58]. The technique is familiar from inclusive semi-leptonic decay into charm quarks, $B \to X_c\ell\nu$ [59–62]. At leading order in this $1/m_b$ expansion the result for the inclusive decay is equal to that for the quark-level process $b \to u\ell\nu$, whose total [63] and differential [64] decay rates are known up next-to-next-to-leading order in QCD. At relative order $1/m_b^2$ only a handful of non-perturbative parameters appear, and recently even for these power corrections the next-to-leading-order QCD corrections have been calculated [65].

When inclusive charmless semi-leptonic decay is considered in experiments, a series of kinematic cuts is required to segregate the charmless decays from the overwhelming amount of charm background. The total rate becomes a partial rate and the convergence of the local OPE tends to be subsequently destroyed. Non-perturbative effects from the Fermi motion of the heavy quark inside the B meson are introduced as the local OPE is replaced by a non-local, shape function OPE. The leading-order contribution in the corresponding $1/m_b$ expansion involves a single non-perturbative shape function [58, 66], which is a function of one light-cone variable. It can be measured from the photon energy spectrum in $B \to X_s\gamma$ [58, 67]. This leading order shape function is universal for all heavy-to-light transitions. Analyses in soft-collinear effective theory have shown that the $1/m_b$ power corrections in this non-local OPE involve a plethora of subleading shape functions beyond tree level, some of which are a function of up to three light-cone variables [68–70], and that the next-to-next-to-leading order QCD corrections to the leading-power decay rate can be substantial [71]. Further theoretical description of the shape functions can be found in Ref. [72–74]. A complete review on inclusive calculations can be found in Ref. [3].

Exclusive predictions focus on a specific hadronic final state resonance, they are complementary to inclusive calculations. The matrix element of the hadronic part can be written as:

$$\langle M(p_M)|\bar{u}\gamma^\mu P_L b|B(p_B)\rangle = \sum T_i^\mu F_i(q^2) \ , \tag{4.1.4}$$

where $q = p_B - p_M$ the four-momentum transfer in the decay, M is the light resonant final state such as π, ρ and ω. T_i are tensorial structures of the involved four-momenta and polarisations in case of vector boson final states, and F_i are form factors. Computing exclusive rates is a challenge due to the form factors because they can not be calculated through perturbation theory in the strong coupling constant and non-perturbative methods are required.

A popular exclusive mode for $|V_{ub}|$ determination is $B \to \pi l \nu$, the hadronic matrix element from Ref. [75] is given as:

$$\begin{aligned}\langle \pi(p_\pi)|\bar{u}\gamma^\mu P_L b|B(p_B)\rangle &= T_1^\mu F_+(q^2) + T_2^\mu F_0(q^2)\\ \text{with } T_1^\mu &= \left((p_B^\mu + p_\pi^\mu) - \frac{m_B^2 - m_\pi^2}{q^2} q^\mu\right)\\ T_2^\mu &= \left(\frac{m_B^2 - m_\pi^2}{q^2} q^\mu\right) \ .\end{aligned} \tag{4.1.5}$$

More exclusive hadronic matrix elements can be found in Ref. [75].

4.2 Status of $|V_{ub}|$

The least known element of the CKM matrix is $|V_{ub}|$, which can be determined at B-factories from semi-leptonic B-decays in the exclusive $B \to \pi \ell \nu$ channel [76–79] as well as from inclusive $B \to X_u \ell \nu$ decays [80–82]. Moreover, it can be tested at the

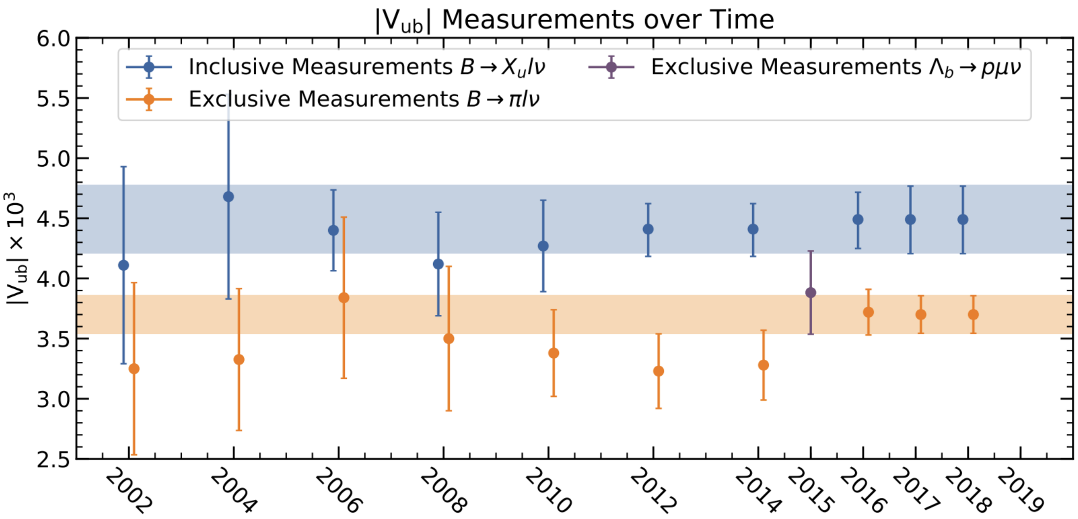

Figure 4.2: $|V_{ub}|$ values extracted from inclusive and exclusive measurements. Figure adopted from Ref. [85].

LHCb experiment in $\Lambda_b \to \rho\mu\nu_\mu$ decays [83]. The current average value of inclusive and exclusive measurements is $|V_{ub}| = (3.82 \pm 0.24) \times 10^{-3}$ [84]. However, there is a long-standing 3σ tension between them, making the determination of $|V_{ub}|$ in the inclusive mode an exciting future measurement for Belle II. The evolution of the $|V_{ub}|$ measurements over the last decade in shown in Fig. 4.2.

From the experimental standpoint, the large background from charmed final states precludes a straightforward measurement of the total inclusive $B \to X_u \ell\nu$ decay rate. The traditional approach to inclusive $|V_{ub}|$ measurements has thus been to make kinematic cuts to restrict measurements in phase-space regions which, neglecting detector effects, are free from charm background. Examples of such cuts are $M_X < m_D$, where M_X is the invariant mass of the hadronic final state X and m_D is the D-meson mass, or $P_+ < m_D^2/m_B$, where $P_+ = E_X - |\vec{P}_X|$ is the energy-momentum difference of the hadronic final state and m_B is the B-meson mass. A technical challenge from the experiments is that the detector effects cause the charm background to populate these theoretically charm-free phase-space regions (see Fig. 4.3 below). Combining this with the task of acquiring a non-trivial separation between signal and background for these restrictive kinematic cuts means that the theoretical description of the partial $B \to X_u \ell\nu$ decay rates becomes considerably more involved. As experiments improve in precision, the partial decay rates require phase-space cuts limit into the non-perturbative shape function region, where the hadronic final state is a collimated jet whose energy is much larger than its invariant mass.

Phenomenologically, several theoretical approaches to partial $B \to X_u \ell\nu$ decay rates are used in $|V_{ub}|$ extractions, going under the acronyms ADFR [86], BLNP [73, 87], DGE [88] and GGOU [74]. These differ in the treatment of QCD effects in the shape function region, but all reduce to the conventional, local OPE results if the kinematic cuts do not introduce new scales which are parametrically much smaller than the b-quark mass. Given the complicated structure of the factorisation theorems and the debate over the precise nature of the shape-function OPE, it is clearly desirable to extend measurements over as large a region of phase space as possible, such that the theoretically clean local OPE results can be applied.

Multivariate analysis techniques based on machine learning (ML) are ideally suited for accessing the $B \to X_u \ell\nu$ decays in regions dominated by the $B \to X_c \ell\nu$ background, while still achieving good signal-to-background ratios. From the ML perspective, the challenge is to build a classifier between signal ($B \to X_u \ell\nu$) and background ($B \to X_c \ell\nu$ and other decays). The first example of such a ML approach to $|V_{ub}|$ determinations was the Belle analysis of Ref. [80]. It used a boosted decision tree (BDT) based classifier taking various high-level kinematic and global features as input and gave a result for the partial decay rate with the single restriction that the charged lepton carries momentum greater than 1 GeV in the B-meson rest frame. Thereby, it samples more than 90 % of the inclusive $B \to X_u \ell\nu$ phase space such that a theoretical description based on the local OPE is applicable. A potential criticism is that such a classifier needs to be trained on Monte Carlo (MC) samples of signal and background events, and is thus especially susceptible to systematic uncertainties based on the kinematic modelling of the signal. A possible approach to evading this criticism was presented in the reanalysis of the Belle data in Ref. [82], where kinematic properties were not included as input features in the BDT classifier; this approach uses the BDT as an additional event selection filter, the result can be used to enhance the signal-to-background ratio to a level which permits a binned one- and two-dimensional likelihood analyses of the kinematic features of the signal and background after event selection.

The rest of this chapter is a systematic study on the use of ML-based classifiers for inclusive $|V_{ub}|$ analyses. There are two main aspects to this study. First, we explore the use of neural networks (NNs) as an alternative ML algorithm to BDTs. While BDTs typically work best when given a small set of carefully engineered, high-level features such as the hadronic invariant mass, NNs can take high-dimensional set of low-level features characterising the event (such as the four-momenta of the final-state particles) as input and use it to learn an optimal way to classify signal and background.[1]

Second, we study in detail the inclusivity of the classifiers and their sensitivity not only to the set of input features chosen, but also to the event generator used producing the training data. In particular, while present $|V_{ub}|$ analyses rely on the generator EvtGen [92], in this paper we compare results using combinations of Sherpa [93] and EvtGen event samples, which differ very little in their description of the $B \to X_c \ell \nu$ background but much more so in the description of the $B \to X_u \ell \nu$ signal.

4.3 Event generation

Our analysis aims at distinguishing $B \to X_u \ell \nu$ signal events from the ~ 50 times larger background induced by the CKM-favoured $B \to X_c \ell \nu$ process. Other background contributions from continuum and combinatorial backgrounds are neglected. The training and test samples of the signal and background events for our ML analyses are produced using MC event generators. In this section we explain our simulation set-up and explore characteristics of the signal and background before

[1] For some discussions on the benefits of using low-level features rather than expert engineered high-level input features only, see e.g. Refs. [89, 90] or the ML review [91].

and after a detector simulation. We also compare MC samples produced with the default generator for B-physics analyses, EvtGen-v01.07.00 [92], with those from Sherpa-v2.2.8 [93].

4.3.1 Monte Carlo samples and event selection

Our event samples are generated at SuperKEKB/Belle II beam energies of $4\,\mathrm{GeV}$ and $7\,\mathrm{GeV}$ or, equivalently, an $\Upsilon(4S)$ resonance with a four-momentum of $p_{\Upsilon(4S)} = (11, 0, 0, 3)\,\mathrm{GeV}$.

For the EvtGen sample, we generate signal and background events with the default run card. For the $B \to X_u \ell \nu$ signal we use the built-in hybrid model for combining resonant and non-resonant modes, with the default input values $m_b = 4.8\,\mathrm{GeV}$ for the b-quark mass, $a = 1.29$ for the Fermi motion parameter and $\alpha_s(m_b) = 0.22$ for the strong coupling at the b-quark mass. The fragmentation of the X_u system into final-state hadrons is performed by Pythia8 [94, 95], and final state QED radiation is performed by Photos [96, 97]. In the Sherpa simulations, we make use of the standard run card for B-hadron pair production on the $\Upsilon(4S)$ pole and use the Sherpa default settings for fragmentation.

In both cases, our baseline event selection process is based on Ref. [80]. We select events with one fully hadronically decaying B meson on the tagging side (B_{tag}), and require the other B meson on the signal side (B_{sig}) to decay semi-leptonically to an electron or muon with $p_\ell^* > 1.0\,\mathrm{GeV}$, where p_ℓ^* is the magnitude of the electron or muon momentum in the B-meson rest frame.

4.3.2 Detector effects

In order to mimic detector effects, we pass our MC data through an in-house detector simulation described in Appendix A.1. In that appendix we also show some validation plots comparing our MC samples with those produced by the Belle collaboration (see Fig. A.1).

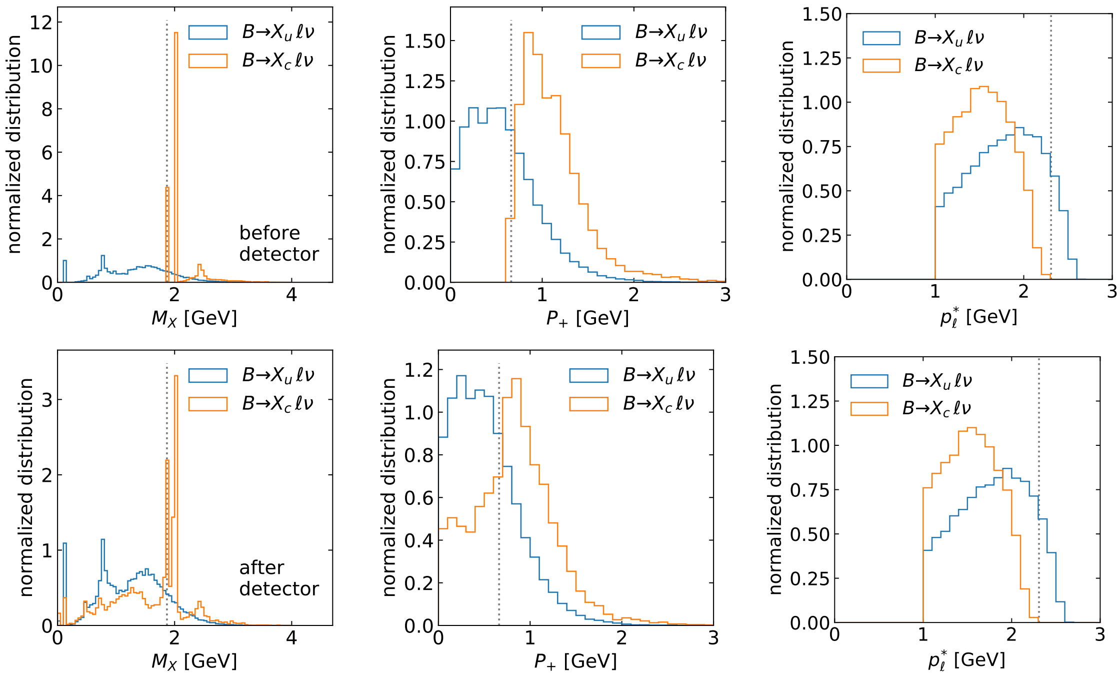

Figure 4.3: EvtGen hadronic mass distribution M_X, energy-momentum difference P_+ and lepton momentum in B-meson rest frame p_ℓ^* before (top) and after detector simulation (bottom). The gray lines highlight the boundaries of the theoretically background-free regions.

Our detector simulation includes detector efficiencies and mistagging for particles on the signal side; it does not take into account that decay products from the tag side can be incorrectly assigned as signal-side particles. While this in-house detector simulation is too simplified to create completely realistic event samples, it does show good agreement with MC results from the Belle collaboration, and can be considered sufficient for the purpose of the qualitative studies performed in this chapter.

In Fig. 4.3, we show normalized distributions of signal and background events in the EvtGen MC sample before and after detector simulation for three kinematic variables: the hadronic invariant mass M_X, the energy-momentum difference P_+, and the lepton momentum in the B-meson rest-frame p_ℓ^*. The distributions of M_X and P_+, which are based on multiple final-state particles and are therefore subject to a cumulative effect from detector inefficiencies and mistagging, are clearly strongly affected by detector effects. In the low-M_X and low-P_+ regions, detector effects cause the charm background to populate even the theoretically inaccessible phase-space

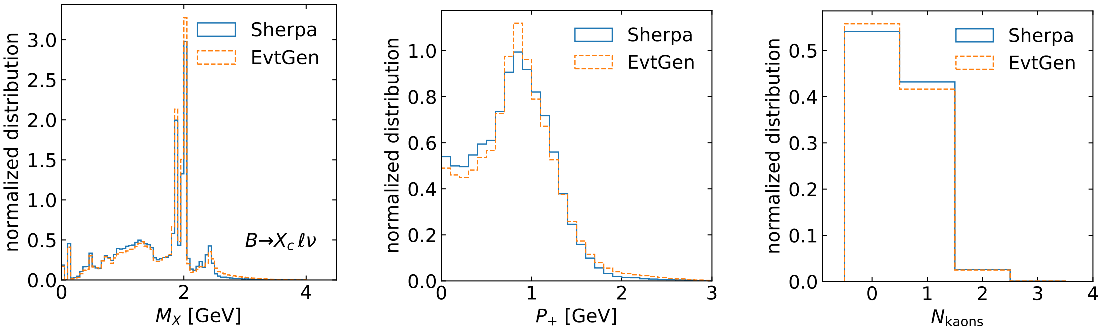

Figure 4.4: High-level features of $B \to X_c \ell \nu$ events generated with EvtGen and Sherpa.

regions $M_X < m_D$ and $P_+ < m_D^2/m_B$. The lepton momentum, on the other hand, can be determined quite precisely and detector effects have only a marginal effect.[1] The contamination shown in these plots make clear that kinematic cuts on their own are insufficient for an efficient signal and background separation after detector effects. We will list a full set of distinguishing features of the signal used in our ML analysis in Section 4.4.1.

4.3.3 EvtGen vs. Sherpa

While EvtGen and Sherpa follow the same general principle in modelling resonant contributions, they differ in the treatment of the non-resonant modes (shape-function regions). In this section we highlight the effects of these modelling choices on distributions of the signal and background.

In Fig. 4.4, we compare distributions for the $B \to X_c \ell \nu$ background. In addition to the kinematic features M_X and P_+, we also show the number of kaons N_{kaons} in the event. Given that inclusive semi-leptonic decays into charm are nearly saturated by a small number of resonant contributions, it is not surprising that the EvtGen and Sherpa results show a close agreement. Minor differences, for instance the number of kaons, are caused by small discrepancies in the assumed branching ratios for high-mass X_c resonances as well as the different hadronisation modelling in Pythia8

[1] This would also be the case in a more realistic simulation, as long as the four-momentum of the tag-side B meson, which determines the boost to the signal B-meson rest frame, is well reconstructed.

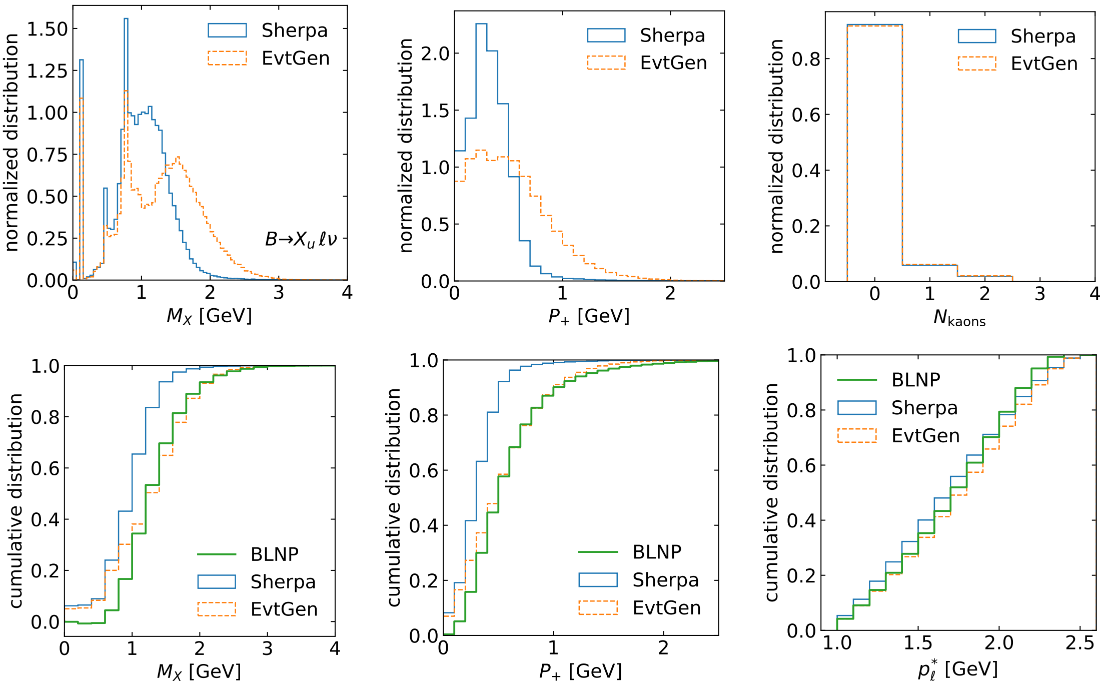

Figure 4.5: Upper panel: Comparison of EvtGen and Sherpa high-level features for $B \to X_u \ell \nu$ signal events. Lower panel: Cumulative sum of the differential distributions M_X, P_+ and p_ℓ^* in EvtGen and Sherpa, compared to BLNP prediction.

and Sherpa.

The analogous distributions for the $B \to X_u \ell \nu$ signal are shown in the upper panel of Fig. 4.5. There are clear differences between the EvtGen and Sherpa distributions of kinematic features such as the M_X distribution, which are caused by the different treatment of the non-resonant modes. In EvtGen, the built-in hybrid model describes the non-resonant decay modes at leading order in the heavy-quark expansion using the DeFazio-Neubert (DFN) model [72], including a non-perturbative shape function to describe the Fermi motion of the b quark inside the B meson. The non-resonant contribution is modelled such that the M_X distribution for the sum of the resonant and non-resonant contributions matches the distribution predicted by the DFN model. This is achieved through a bin-by-bin re-weighting of the non-resonant modes.

In Sherpa the non-resonant signal decay modes are modelled by parton showering and hadronizing the leading-order partonic decay. Non-perturbative shape-function effects characterising the low-M_X region are not taken into account, and no re-weighting of the events is performed to match state-of-the-art theory calculations.

Comparing these two different approaches for the signal modelling in Fig. 4.5, we find that, on the one hand, the EvtGen results have a non-physical bump in the 1.5 GeV region, which is an artefact of the bin-by-bin re-weighting to match the DFN results. The Sherpa distributions do not share this characteristic, since the non-resonant events are instead obtained by excluding resonant events from the parton shower. On the other hand, the current implementation of the Sherpa parton shower model also produces a smaller proportion of the non-resonant signal contribution and generates fewer events in the high-M_X and -P_+ regions compared to EvtGen, which is precisely the region where the inclusive QCD predictions should be reliable. We further highlight this in the lower panel of Fig. 4.5, where we compare the state-of-the-art OPE results from the BLNP approach [73] with EvtGen and Sherpa results at the level of cumulative distributions. Overall, the agreement between the EvtGen-generated distributions with the BLNP predictions is stronger, which is not surprising since the underlying inclusive modelling comes from the OPE-based DFN result.

Clearly, the $B \to X_u \ell \nu$ modelling in Sherpa needs a more sophisticated matching of the non-resonant, parton shower contributions with (shape-function) OPE results before being used in $|V_{ub}|$ extractions by experiments. For this reason, we use EvtGen in the following section when studying the performance of ML-based classifiers, in spite of its own deficiencies in the low and intermediate invariant mass regions. However, for the purposes of the study, the present situation allows us to study an interesting question: how sensitive to the MC data used in the training process are ML approaches to $|V_{ub}|$ extractions? This is the subject of Section 4.5.

4.4 BDTs vs NNs

In this section we give a systematic analysis of signal vs. background event classification using BDTs and neural networks. We use Bayesian neural networks (BNNs)[1], which have been argued to deliver stable results and avoid overfitting [54]. The details of the architecture for the BDTs and BNNs used in our study can be found in Appendix A.2, along with a breakdown of data used in the training and testing procedure. We describe the input features to the ML algorithms in Section 4.4.1, and then move on to the results in Section 4.4.2. Note that the metrics used in evaluating their performance have been described Section 3.4. Throughout this section we use EvtGen to generate the training and testing samples.

4.4.1 Input features

The features used in our multivariate analysis break into two sets. One is based on physical high-level features such as invariant masses and the number of final-state particles of a specific type, e.g. the number of kaons or slow pions, and the other is based on low-level features, i.e. single particle properties. In particular, the low and high-level features are:

- **low level**

$$p_{B_{\text{tag}}},\, Q_{B_{\text{tag}}},\, p_i,\, \text{ID}_i,\, Q_i \quad i \in \text{top 10 most energetic particles.} \tag{4.4.1}$$

- **high level**

$$\begin{gathered} q^2, \quad M_X, \quad P_+, \quad p^*_\ell, \quad N_\ell, \quad N_{K^\pm}, \quad N_{K^0}, \quad N_{\text{hadron}}, \quad M^2_{\text{miss}}, \quad Q_{\text{tot}}, \\ N_{\pi^0_{\text{slow}}}, \quad N_{\pi^\pm_{\text{slow}}}, \quad M^2_{\text{miss},\,D^*}(\pi^0_{\text{slow}}), \quad M^2_{\text{miss},\,D^*}(\pi^\pm_{\text{slow}}). \end{gathered} \tag{4.4.2}$$

[1] BNN and NN are used interchangeably throughout this chapter but they both mean Bayesian neural network.

The low-level features include, first off, the four-momentum $p_{B_{\text{tag}}}$ and charge $Q_{B_{\text{tag}}}$ of the tagged B meson. In addition, we pick out the 10 most energetic (as measured in the lab frame) detected final-state particles, label them with an index $i = 1, \ldots 10$, and use as features the lab frame four-momenta p_i, the charge Q_i and the identity ID_i of these particles. Events with less than 10 detected final-state particles have the corresponding particle features filled in with zeros.

The high-level features are defined as follows. The four-momentum transfer squared is $q^2 = (p_B - p_X)^2$. N_ℓ denotes the number of leptons, which can only be greater than one if the secondary leptons have momenta smaller than 1 GeV. Since the $B \to X_u \ell \nu$ signal is very unlikely to contain secondary leptons, this feature can be used to suppress the background, see the left panel of Fig. 4.6. $N_{K^\pm}$ and N_{K^0} denote the number of charged and neutral kaons, respectively, where neutral kaons K_S^0 are reconstructed from charged pions with an invariant mass in the range $m_{\pi^+\pi^-} \in [0.490, 0.505]\,\text{GeV}$. Kaons are frequently produced in D-meson decays and their presence hence indicates a $B \to X_c \ell \nu$ background event, see the central panel of Fig. 4.6. The number of final-state particles resulting from the hadron decay N_{hadron} is typically larger for hadrons with a higher mass such as the background D mesons. The missing mass squared M_{miss}^2, defined as the square of the missing momentum $p_{\text{miss}} = p_{\text{sig}} - p_X - p_\ell$, where $p_{\text{sig}} = p_{\Upsilon(4S)} - p_{B\text{tag}}$ is the reconstructed momentum of the signal-side hadron, would always be compatible with zero without detector effects. For background events, which as discussed above have a higher final-state particle multiplicity, the probability of misidentifying a final-state particle is higher resulting in positive values of the missing mass squared, see the right panel of Fig. 4.6. The total charge Q_{tot} of all particles in the event, on both the signal and the tag side, is also subject to detector effects. It will only be non-zero for events where charged particles have been missed, which happens more often for the background events due to their larger final-state particle multiplicity. Slow pions, i.e. pions with momentum $|p_\pi| < 220\,\text{MeV}$, can originate from $D^* \to D\pi$ transitions and hence appear more often for the $B \to X_c \ell \nu$ background. We therefore include the number

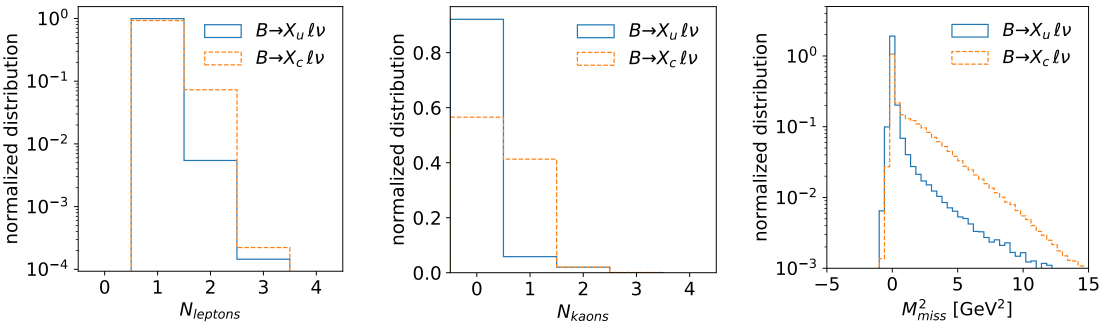

Figure 4.6: High-level features of the EvtGen sample. Number of leptons N_ℓ (left), number of kaons N_{kaons} (middle) and missing mass squared M^2_{miss} (right). Notice the logarithmic scale for some of the distributions.

of neutral and charged slow pions, $N_{\pi^0_{\text{slow}}}$ and $N_{\pi^\pm_{\text{slow}}}$, in our high-level feature set. To test the compatibility of the slow pion with a $D^* \to D\pi$ transition, we further define $M^2_{\text{miss},\,D^*} = (p_{\text{sig}} - p_{D^*} - p_\ell)^2$, where $p_{D^*} = (E_{D^*}, \vec{p}_{D^*})$ with $E_{D^*} = \frac{m_{D^*}}{m_{D^*} - m_D} E_\pi$ and $\vec{p}_{D^*} = \vec{p}_\pi \frac{\sqrt{E^2_{D^*} - m^2_{D^*}}}{|\vec{p}_\pi|}$. In this we have explicitly assumed that the slow pion direction is strongly correlated with the D^* direction. The quantity $M^2_{\text{miss},\,D^*}$ will more likely be peaked at zero for true $D^* \to D\pi$ transitions. Distributions in the high-level input features not shown in Fig. 4.6 are displayed in Appendix A.3 in Fig. A.2.

We have chosen this set of high-level features to mimic the feature selection in the BDT analyses performed by Belle in Refs. [80, 82]. Some differences with respect to the sets used in those papers arise, because we do not have access to all experimental features in our simplified detector simulation, for instance features related to the quality of the signal reconstruction.

4.4.2 BDT and NN performance on different levels of input features

We first contrast the performance of the BDT and NN on signal vs. background classification using different levels of input features. We consider three scenarios:

(i) using only the low-level features in Eq. 4.4.1

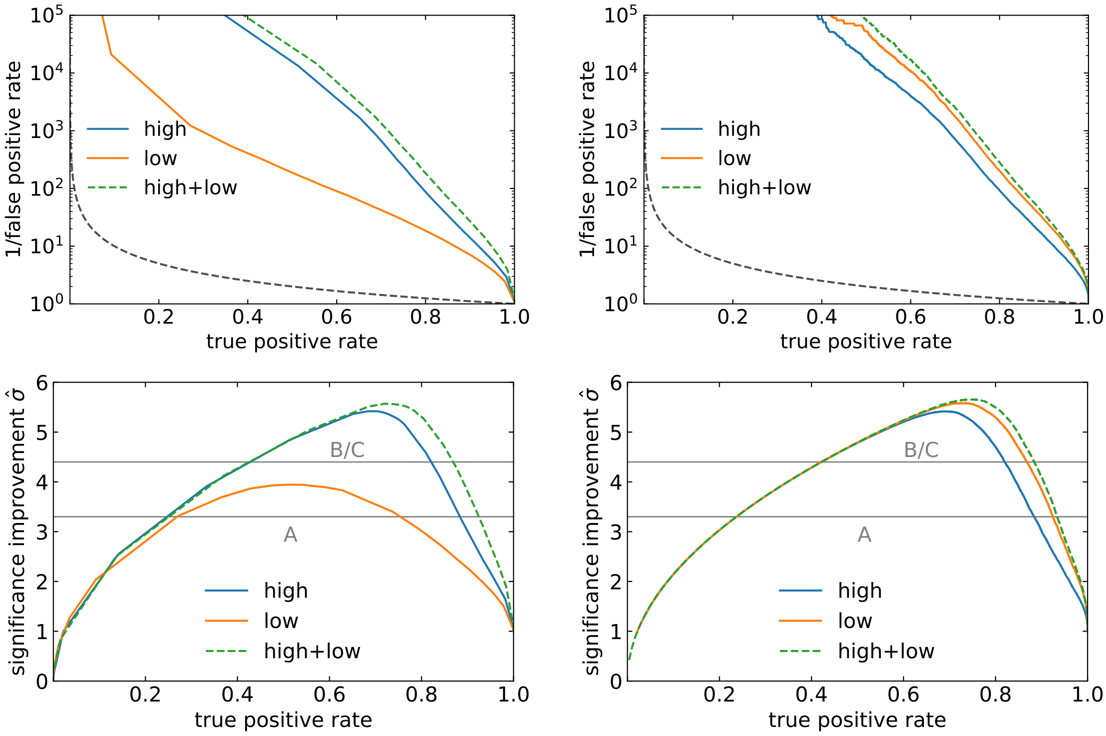

Figure 4.7: ROC (top) and SIC curves (bottom) for BDT (left) and NN (right) for different levels of input features, trained and tested on EvtGen data with a physical ratio of signal-to-background events in the test set. The dashed lines in the upper panel are ROC curves for the case of no separation. As a reference, the gray lines in the bottom panel show the significance improvement from the three cut-and-count scenarios in Eq. 4.4.5. A: $M_X < m_D$, B: $M_X < 1.5\,\text{GeV}$, C: $P_+ < m_D^2/m_B$.

(ii) using only the high-level features in Eq. 4.4.2

(iii) using a combination of these low- and high-level features.

The ROC and SIC curves for the BDT and NN analyses using these input feature scenarios are shown in Fig. 4.7.

As expected, the BDT performs well on high-level input features, the most important features being the number of kaons, number of leptons, the hadronic invariant mass M_X, hadron multiplicity and the missing mass squared M^2_{miss}. However, it performs poorly when trained only with low-level features, indicating that it cannot use them to construct additional non-linear features such as invariant masses. Using a combination of low- and high-level features slightly improves the BDT performance compared to high-level only. We have explicitly checked that this performance increase results almost entirely from adding the particle energies. The particle three-momenta, on the other hand, do not seem to contain additional usable information for the BDT.

The situation for the NN is very different. It performs slightly better when trained only on low-level features than it does when trained only on high-level features. This indicates that, as expected, it is able to learn new and efficient discriminating features from the low-level inputs. Training on a combination of low plus high-level inputs very marginally improves its performance compared to low-level only (mainly due to the inclusion of M_X as a feature), showing that the NN has learnt the most important high-level features on its own.

The maximum of the SIC curves is reached for a cut on the classifier output of $\zeta_{\text{cut}} \approx 0.97$, which corresponds to a signal acceptance, or true positive rate $\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$, of approximately 75 %. Explicitly, we find the following values for the maximum significance improvement and the AUC for a BDT or NN trained and tested on a combination of high and low-level features from the EvtGen data:

$$\begin{aligned} \text{AUC} &= 0.981, \quad \hat{\sigma} = 5.59 \quad \text{BDT} \\ \text{AUC} &= 0.986, \quad \hat{\sigma} = 5.67 \quad \text{NN}\,. \end{aligned} \tag{4.4.3}$$

The AUC and $\hat{\sigma}$ for the NN is only about 2 % better than the BDT approach. Training on high-level features only puts the NN on equal footing with the BDT – in fact, we find that they reach the exact same significance improvement, which is $\hat{\sigma} = 5.42$. The very small loss of performance compared to the Eq. 4.4.3 indicates that the high-level features are well chosen for a discrimination of signal and background, containing (almost) the full relevant information that the NN can learn from the low-level features and given architecture.

It is interesting to contrast the significance improvements using the BDT and NN with those obtained from a typical cut-and-count analysis based on the cuts provided in Ref. [81]. With the minimal requirement of having exactly one lepton, a total charge of zero, a veto on kaons and a low missing mass squared,

$$N_\ell = 1, \quad Q_{\text{tot}} = 0, \quad N_{\text{kaons}} = 0, \quad M^2_{\text{miss}} < 0.5\,\text{GeV}^2\,, \tag{4.4.4}$$

we obtain a significance improvement of $\hat{\sigma} = 1.9$. If in addition to theses cuts we select a theoretically background-free region, we find[1]

$$\hat{\sigma}(M_X < m_D) = 3.3, \qquad \hat{\sigma}(M_X < 1.5\,\text{GeV}) = 4.4, \qquad \hat{\sigma}(P_+ < m_D^2/m_B) = 4.4\,. \tag{4.4.5}$$

Comparing the significance values Eq. 4.4.5 with those from the BDT and NN analysis in Eq. 4.4.3, we see that the ML approaches clearly outperform the cut-and-count analyses.

4.5 Inclusivity of ML approaches

A main motivation for the application of ML techniques to $|V_{ub}|$ determinations is to widen the experimentally accessible fiducial region to a level of inclusivity where the theoretically clean, local OPE is unambiguously applicable. This amounts to two conditions on the measured X_u final state: first, that it is not subject to severe kinematic cuts (in which case the shape-function OPE would apply), and second, that it contains a sufficiently broad sample of exclusive hadronic final states in a given kinematic region (such that quark-gluon duality applies). A concern in supervised ML approaches is that the classifiers will overuse either inclusive kinematic properties or IR unsafe hadron-level properties of the final state, thereby limiting the signal output to a restricted fiducial region which is very sensitive to MC modelling, regardless of the inclusivity of the input events.

In this section we study the inclusivity of the signal acceptance in ML approaches to event classification. As the inclusivity depends crucially on the input features used

[1] We consider the cut scenario $M_X < 1.5\,\text{GeV}$ in addition to $M_X < m_D$ to account for the fact that the background will dominantly populate the region slightly below m_D due to detector effects, see Fig. 4.3.

in the ML classifier, we consider two scenarios:

- **NN$_{\mathbf{tight}}$**: a BNN using as input both the low and high-level features listed in Eq. 4.4.1 and Eq. 4.4.2, respectively. This is a more sophisticated implementation of the basic approach of Ref. [80], and its classification power was explored in Section 4.4.2.

- **NN$_{\mathbf{loose}}$**: a BNN using as input the high-level features listed in Eq. 4.4.2, but *excluding* the kinematic features M_X, P_+, q^2 and p_ℓ^*. This is a proxy for the BDT used in the recent reanalysis of Belle data [82].

In both cases the classifier threshold is chosen to maximize the significance of the accepted event set. Obviously NN$_{\text{loose}}$, which intentionally excludes discriminating kinematic features of the signal and background, will not lead to the same signal purity as NN$_{\text{tight}}$. In our analysis NN$_{\text{tight}}$ reaches a signal-over-background ratio of $S/B \sim 13$, while for NN$_{\text{loose}}$ $S/B \sim 0.3$ such that the background contribution is still dominant even after event selection by the NN. In this latter case it is thus essential to perform a binned one- and two-dimensional likelihood analyses of the kinematic features of the signal and background after event selection by the NN$_{\text{loose}}$, as was done in Ref. [82]; this procedure can be useful for NN$_{\text{tight}}$ as well, even though the S/B ratio is much higher.

A main focus of our study is how changes of the testing and training data affect the inclusivity of the ML analyses. Testing and training the NNs on differently modelled event sets provides a good test for overtraining and gives insight into how well the classifier might perform when applied to real-world events, which are not expected to show perfect agreement with MC data. The existing ML-based Belle analyses [80,82] estimate uncertainties stemming from input data modelling by testing on samples produced with different parameter choices within the EvtGen framework while fixing the ML configuration. Here we explore the alternative method of using a fundamentally different MC-event generation framework, namely Sherpa. In this section we train all NNs on EvtGen and then study their classification properties on both Sherpa and EvtGen data; in Appendix A.4 we show equivalent results when the NNs are trained instead on Sherpa data. All MC samples used in testing the NNs, whether generated by Sherpa or EvtGen, contain the same ratio of signal to background events after detector simulation.

We compare the inclusivity of the two NN setups in two main ways. In Section 4.5.1, we study the inclusivity in kinematic phase space, and in Section 4.5.2 we focus on inclusivity in the available hadronic final states. In the latter section we also study sensitivity to changes of hadronisation parameters within the EvtGen framework. In addition to physics properties, we show the effect of ML training sample weights in Section 4.5.3 as an alternative method to boost inclusivity.

4.5.1 Inclusivity in kinematics

We illustrate the salient features of event selection by $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{loose}}$ as a function of M_X, q^2, and p_ℓ^* in Fig. 4.8. The binning of the kinematic variables matches that used in the fitting procedure of the recent $|V_{ub}|$ extractions in Ref. [82]:

$$\begin{aligned} M_X &= [0,\ 1.5,\ 1.9,\ 2.5,\ 3.1,\ 4.0]\,\mathrm{GeV}\,, \\ q^2 &= [0,\ 2,\ 4,\ 6,\ 8,\ 10,\ 12,\ 14,\ 26]\,\mathrm{GeV}^2\,, \\ p_\ell^* &= 15 \text{ equidist. bins in } [1,\ 2.5]\,\mathrm{GeV}\ \&\ \ [2.5,\ 2.7]\,\mathrm{GeV}\,. \end{aligned} \tag{4.5.1}$$

In all cases, the bins are sufficiently wide that the results can be compared with predictions from the (shape-function) OPE, after correcting for acceptances and detector effects. Each plot in the figure shows the following three results for the indicated MC event sample: the detector-level signal distributions and the total number of events (TP+FP) accepted by the given NN (upper panels), and the signal acceptance of the NN (lower panels), all normalized to the number of detector-level signal events. The left (right) column uses $\mathrm{NN}_{\mathrm{tight}}$ ($\mathrm{NN}_{\mathrm{loose}}$). The NNs are trained on EvtGen data, and then tested on both EvtGen and Sherpa data. For $\mathrm{NN}_{\mathrm{loose}}$, we also display the background acceptance in the lower panels, using the scale for the y-axis displayed on the right of the plots. The background acceptance for $\mathrm{NN}_{\mathrm{tight}}$ is negligible across phase space and is thus not shown.

The figure highlights an inevitable fact – since $\mathrm{NN}_{\mathrm{tight}}$ uses kinematic features to discriminate between the signal and background, its acceptance is kinematics dependent. The acceptance is higher in the theoretically background-free regions of low M_X, high q^2, and high p_ℓ^*, and lower in regions where the charm background is large.

It is interesting and important to study the MC-data dependence of the signal acceptance in these two regions, and connect it to kinematic modelling uncertainties in the MCs. Take for example the results as a function of M_X in the top left of the figure. In the $0 < M_X < 1.5$ GeV bin, the EvtGen and Sherpa modelling of the $b \to u$ signal differ dramatically, with far more events in the Sherpa sample, and also a very different shape as seen in the finely binned distributions shown in Fig. 4.5. This is not entirely unreasonable, as the details of the low-M_X distributions depend on the method for matching resonant and non-resonant modes, and even the integrated distribution over the entire bin depends on the exact implementation of the shape-function OPE. However, the MC-dependence of the signal distribution in this theoretically intricate region does not propagate into the signal acceptance of $\mathrm{NN}_{\mathrm{tight}}$, which is essentially MC-independent.

Contrast this with the high-M_X region, especially in the bins above 1.9 GeV where the charm background is large. In this case, the marked difference in the shapes of the EvtGen and Sherpa signals as a function of M_X does lead to noticeably different signal acceptances. On the other hand, kinematic distributions in the high-M_X region where this becomes most significant are reliably calculable within the local OPE (before detector effects), so the MC-dependence can be viewed as an improvable deficiency in the current implementation of Sherpa, which does not perform a matching with first-principle predictions as described in Section 4.3.3, rather than as an irreducible kinematic modelling uncertainty. One would therefore expect a reasonable MC uncertainty associated with extrapolating the accepted events to the full fiducial region, although this deserves careful quantitative study in actual experimental analyses.

Similar qualitative comments hold for the p_ℓ^* and q^2 distributions – the signal acceptances are essentially MC-independent in the highest bins, where kinematic modelling dependence due to non-perturbative shape-function effects is expected to be significant, but then start to become MC-dependent in the lower bins, where the local OPE is applicable. On the other hand, the acceptances are somewhat flatter in these variables than in M_X, never dropping below 60% in any of the bins.

The exclusion of kinematic input features from NN_loose leads to a different qualitative picture of event acceptance compared to NN_tight. The right-hand side of Figure 4.8 shows that its signal acceptance as a function of M_X is considerably flatter, remaining large at and above the m_D resonance, although at the price of rejecting far less background. In total, NN_loose also accepts less of the signal. Whereas NN_tight accepts 75% (85%) of the EvtGen (Sherpa) signal, the corresponding numbers for NN_loose are 61% (53%) at the value of the threshold classifier which optimizes the significance improvement. For the q^2 and p_ℓ^* distributions the acceptances of NN_loose are only moderately flatter than NN_tight, if at all. The signal acceptances of NN_loose are reasonably independent of the MC testing data across the kinematic phase space. However, unlike NN_tight, noticeable differences can be seen in the lowest M_X and highest q^2 and p_ℓ^* bins, where shape-function effects and kinematic modelling are expected to be most important. The background acceptance of NN_loose is relatively flat at high M_X and low p_ℓ^*, but not at low q^2. Moreover, in the lowest M_X bins as well as the high-q^2 region the background is largely excluded; these regions correlate with a large missing mass squared.

These observations show that MC-dependence of the acceptances of a given NN is subtle – avoiding sensitivity to kinematic modelling by excluding kinematic features is not always possible. As a further illustration, consider a BNN, NN_binned, taking as input the following features

$$\begin{gathered} Q_{B_{\mathrm{tag}}}, \quad \mathrm{ID}_i, \quad Q_i, \quad [q^2]_{\mathrm{binned}}, \quad [M_X]_{\mathrm{binned}}, \quad [p^*_\ell]_{\mathrm{binned}}, \quad N_\ell, \quad N_{K^\pm}, \quad N_{K^0}, \\ N_{\mathrm{hadron}}, \quad M^2_{\mathrm{miss}}, \quad Q_{\mathrm{tot}}, \quad N_{\pi^0_{\mathrm{slow}}}, \quad N_{\pi^\pm_{\mathrm{slow}}}, \quad M^2_{\mathrm{miss},\,D^*}(\pi^0_{\mathrm{slow}}), \quad M^2_{\mathrm{miss},\,D^*}(\pi^\pm_{\mathrm{slow}}). \end{gathered} \tag{4.5.2}$$

$\mathrm{NN}_{\mathrm{binned}}$ is the same as $\mathrm{NN}_{\mathrm{tight}}$, except that particle 4-momenta are excluded[1], and the high-level kinematic features are defined in the bins

$$\begin{aligned} M_X &= [0,\ 1.4,\ 1.6,\ 1.8,\ 2,\ 2.5,\ 3,\ 3.5]\,\mathrm{GeV} \\ p^*_\ell &= [1,\ 1.25,\ 1.5,\ 1.75,\ 2,\ 2.25,\ 3]\,\mathrm{GeV} \\ q^2 &= [0,\ 2.5,\ 5,\ 7.5,\ 10,\ 12.5,\ 15,\ 20,\ 25]\,\mathrm{GeV}^2\,. \end{aligned} \tag{4.5.3}$$

This binning matches that used in the construction of the hybrid Monte Carlo implemented within EvtGen in Ref. [82], and is sufficiently wide that fully inclusive distributions within these bins are accessible to the (shape-function) OPE. In other words, unlike $\mathrm{NN}_{\mathrm{tight}}$, this set-up is blind to the heavily model-dependent point-by-point distributions of the hybrid Monte Carlo in the low M_X and high p_ℓ and q^2 region, at least as far as the explicit input features are concerned.

In Fig. 4.9 we compare the acceptances of $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{binned}}$ as a function of kinematic variables, using the same binning as in Fig. 4.8. Examining the figure shows that the MC-dependence of the $\mathrm{NN}_{\mathrm{binned}}$ acceptances are not reduced compared to $\mathrm{NN}_{\mathrm{tight}}$, and they depend more strongly on the kinematic variables. In particular, when viewed as a function of M_X, $\mathrm{NN}_{\mathrm{binned}}$ shows a considerable drop in classification power in the higher bins, where kinematic modelling uncertainties are expected to be best under control as long as the hybrid Monte Carlo is matched to OPE predictions. Moreover, the maximal significance improvement $\hat{\sigma}$ drops: when tested on EvtGen data $\mathrm{NN}_{\mathrm{tight}}$ has $\hat{\sigma} = 5.67$ while $\mathrm{NN}_{\mathrm{binned}}$ has $\hat{\sigma} = 5.46$. It is thus far from clear that using a set-up such as $\mathrm{NN}_{\mathrm{binned}}$ would lead to a reduced theory uncertainty in $|V_{ub}|$

[1] The high-level features for $\mathrm{NN}_{\mathrm{binned}}$ also differ from $\mathrm{NN}_{\mathrm{tight}}$ in that P_+ is included in the latter case but not the former. We verified that adding or taking it away from makes a negligible numerical difference.

extractions compared to $\mathrm{NN}_{\mathrm{tight}}$, even though its explicit kinematic input features can be calculated within the (shape-function) OPE.

4.5.2 Inclusivity in hadronic final states

We now shift our focus to inclusivity in properties of the final-state X_u system which appear only after fragmentation into hadrons. Such features are by definition inaccessible to OPE-based QCD calculations, which rely on a sum over hadronic final states in order for quark-gluon duality to apply.

In Fig. 4.10 we display the same information as in Fig. 4.8, but this time as a function of the number of kaons and total charge in the event. The number of kaons is an explicit probe of the flavour structure of the final state, whereas the total charge is closely related to the charged hadron multiplicity (see the discussion after Eq. (4.4.2) above). Comparing the acceptance of $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{loose}}$, we find that $\mathrm{NN}_{\mathrm{loose}}$ effectively vetos both signal and background events with kaons or a non-zero total charge.[1] Therefore, when performing fits of the kinematic distributions after the $\mathrm{NN}_{\mathrm{loose}}$ analysis, a good understanding of both the signal and the charm background after strict cuts on the hadronic final states is required. $\mathrm{NN}_{\mathrm{tight}}$, on the other hand, accepts a large proportion of events with kaons or a non-zero total charge and is thus more inclusive in (and less dependent on) these hadronisation-model dependent features.

The number of signal events containing kaons in the final state is directly related to the $s\bar{s}$-popping probability γ_s, which determines how often an $s\bar{s}$-pair is produced in the decay of the hadronic X system. It is interesting to further investigate the hadronisation modelling sensitivity of the classifiers $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{loose}}$ resulting from their different kaon acceptances. Since the number of kaons in the background, which is entirely dominated by resonant contributions, is largely unaffected by changes of γ_s, we investigate the sensitivity of the signal acceptance only. We have produced

[1] The small contributions of events with $Q_{\mathrm{tot}} = 2$ to the total number of signal events is negligible.

additional EvtGen test samples with a modified $s\bar{s}$-popping probability in the range $\gamma_s \in [0.1, 0.4]$ and apply $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{loose}}$ to these.[1]

In Fig. 4.11, we display the relative change of the number of TP events as a function of γ_s, taking the Pythia8 default $\gamma_s = 0.217$ [100] as our reference value. As events containing kaons are more likely to be classified as background by the NNs, the number of TP events decreases with an increasing value of γ_s. For $\mathrm{NN}_{\mathrm{loose}}$, which relies more heavily on the number of kaons as a features, the decrease of the signal acceptance is stronger.

We contrast the effect of γ_s on our ML analysis with a simple kaon veto as well as a cut-based approach defined by the cuts listed in Eq. 4.4.4 plus an additional cut $M_X < 1.5\,\mathrm{GeV}$ (tight cuts). The ML approach $\mathrm{NN}_{\mathrm{loose}}$ shows the same influence on γ_s as a kaon veto, as expected from the signal acceptance shown in Fig. 4.10. $\mathrm{NN}_{\mathrm{tight}}$, however, is less disturbed by an increased value of γ_s than its cut-and-count counterpart as it does not apply a stringent veto on kaons in signal events. Overall, our findings highlight the ability of ML approaches to lift the weight from single observables.

4.5.3 Inclusivity boost with ML sample weights

A possible option to increase acceptance from the ML setup is to include sample weights to our training data for signal events in the high-M_X region. Specifically, we have increased the sample weight of signal events with an invariant hadronic mass greater than $M_X > 1.5\,\mathrm{GeV}$ to five, while using a flat sample weight of one for all other signal and background events. Sample weights act as an additional penalty term for the objective function shown in Eq. 3.1.3 such that false classification on weighted events are strongly penalised. It is a common technique when dealing with

[1] The tested γ_s range is chosen to reflect the relatively large uncertainty on γ_s. The TASSO [98] and JADE [99] collaborations have experimentally determined the $s\bar{s}$-popping probability at center-of-mass energies of 12 GeV and 27 GeV to be $\gamma_s = 0.35 \pm 0.05$ and $\gamma_s = 0.27 \pm 0.06$, respectively. The default Pythia8 setting, resulting from a global tune of multiple fragmentation parameters, is $\gamma_s = 0.217$ [100].

imbalanced datasets. We explore this option on NN$_{\mathrm{tight}}$ training and testing with EvtGen data.

In Fig. 4.12, we show the acceptance in terms of M_X on the left and a comparison on the significance improvement on the right between two NN$_{\mathrm{tight}}$, unweighted and weighted training data. Keeping the classifier threshold fixed at 0.97, the sample weights flatten the acceptance as shown in the left plot. However, when we increase the classifier threshold to the value which maximises the significance, in this case 0.99, the acceptance at high M_X drops to the same level as the acceptance of the unweighted sample. Overall, the maximum significance of the ML classification reduces slightly as more background events are accepted in the signal region as shown in Fig. 4.12.

This procedure is not well suited for this study even though it is an imbalanced data problem in experiments, the imbalance does not apply to our training data. The algorithm is able to learn sufficiently well for both classes given a large amount of training data. Therefore, this procedure becomes a balance problem between the significance and the classification threshold.

4.5.4 Discussion

The above results show that conclusions on the inclusivity of NN$_{\mathrm{tight}}$ and NN$_{\mathrm{loose}}$ are based heavily on how one thinks about the issue. If the focus is on a flat coverage of kinematic phase space, especially as a function of M_X, then NN$_{\mathrm{loose}}$, which does not include kinematic features, would be preferable. If on the other hand one wishes to be more inclusive in the sum over exclusive hadronic final states on which quark-gluon duality is based, then NN$_{\mathrm{tight}}$, which accepts more events overall due to its increased discriminating power, is more attractive.

An important thing to keep in mind when considering $|V_{ub}|$ extractions is that in both cases MC modelling is used to extrapolate the signal from the fiducial region singled out by the NN to the partial inclusive branching fractions with a baseline kinematic cut of $p_\ell^* > 1.0\,\mathrm{GeV}$ (with no restrictions on the hadronic decomposition of the X_u final state). For NN$_{\text{tight}}$ this extrapolation is mainly sensitive to the shape of the signal distribution at relatively high M_X, which can reliably be calculated in the local OPE. For NN$_{\text{loose}}$ it is mainly sensitive to non-perturbative phenomena such as the flavour decomposition and multiplicity of the hadronic final state across all kinematics. Given that the extrapolations are sensitive to different effects, it may be wise to pursue both approaches in real-life $|V_{ub}|$ extractions.

It is worth mentioning that the signal acceptance of the kinematics independent "background suppression" BDT used in the recent analysis of Ref. [82] is significantly smaller than that found using NN$_{\text{loose}}$ and our in-house detector simulation, so that the extrapolation from the accepted fiducial region to fully inclusive partial branching fractions with kinematic cuts is correspondingly larger. By the same token, we expect that the acceptance of NN$_{\text{tight}}$ in the high-M_X region would be considerably lower in the full experimental environment, again requiring a larger extrapolation than seen in our simplified set-up.

4.6 Summary

This chapter presented a systematic study on the use of ML techniques in inclusive $|V_{ub}|$ determinations. While our analysis is based on a simplified set-up using an in-house detector simulation and seeking only to separate the $B \to X_u \ell \nu$ signal from the $B \to X_c \ell \nu$ background, it has revealed several important qualitative points.

First, in Section 4.4, we showed that using a deep neural network trained on low-level single-particle features leads to a small performance increase with respect to a BDT analysis based on high-level features of the type used in the Belle analysis [80]. While upgrading such analyses to modern ML standards is certainly worthwhile, the modest performance increase produced by the more sophisticated ML architecture implies that the high-level features used in current BDTs are well-chosen – the most important aspects of discriminating the $b \to u$ signal from the $b \to c$ background can be understood with physicist-engineered observables.

Second, in Section 4.5 we studied the inclusivity of the fiducial region selected by cuts on the classifier output of two types of neural networks: $\mathrm{NN}_{\mathrm{tight}}$, based on input features of both kinematic and hadron-level features of the final states, such as the one just described and used in Ref. [80], and $\mathrm{NN}_{\mathrm{loose}}$, which excludes the kinematic properties and is similar to the BDT used in the recent analysis in Ref. [82]. While the signal acceptance of $\mathrm{NN}_{\mathrm{loose}}$ is fairly flat across the kinematic phase space, it effectively makes hard cuts in hadronic properties of the event such as the number of kaons and the total charge. On the other hand, $\mathrm{NN}_{\mathrm{tight}}$ is significantly more inclusive in the hadronic decomposition of the final state and also in general, but tends to give less weight to kinematic regions where there is a large overlap with the $b \to c$ background. Both of these issues deserve careful consideration when assessing systematic theory uncertainties related to MC extrapolation from the fiducial regions to partial branching fractions that are calculable within the (shape-function) OPE in QCD.

Finally, as the Belle II measurements become systematics dominated, it will be important to pay close attention to the sensitivity of supervised ML approaches to the MC data on which they are trained. We have investigated the influence of a modified $s\bar{s}$-popping probability on the signal acceptance using EvtGen data. A ML approach based on kinematic information, such as $\mathrm{NN}_{\mathrm{tight}}$, is generally less biased by changes of global event parameters. Furthermore, in Section 4.3 we showed results from the multipurpose MC event generator Sherpa in addition to those from EvtGen, which has been the exclusive MC tool for all previous $|V_{ub}|$ analyses, and in Section 4.5 we discussed features appearing when the NNs were trained and tested on event sets produced by different MCs. While Sherpa needs optimisation in matching with OPE-based theory predictions before it can be used in experimental analyses, investigating the stability of ML approaches against MCs whose modelling is based on different theory assumptions can provide a powerful stress-test on MC uncertainties, beyond the current practice of exploring modifications within EvtGen.

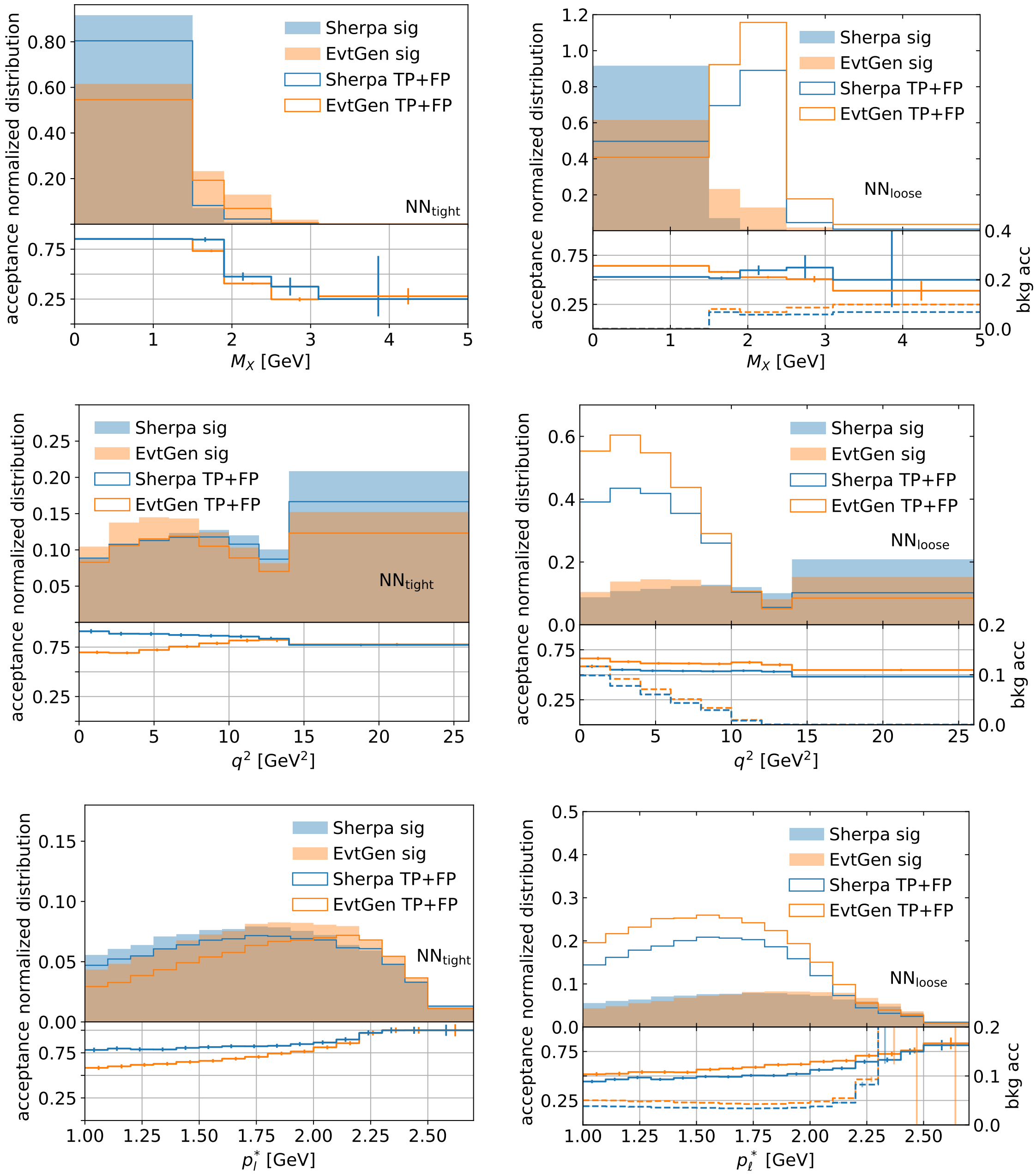

Figure 4.8: Distributions and signal acceptance of Sherpa and EvtGen Monte Carlo data as functions of M_X, q^2, and p_ℓ^* for $\mathrm{NN}_{\mathrm{tight}}$ (left) and $\mathrm{NN}_{\mathrm{loose}}$ (right), trained on EvtGen data. The distributions in the upper panels of each plot are normalized to the total number of signal events. For $\mathrm{NN}_{\mathrm{loose}}$ the dashed lines in the lower panels show the background acceptance, using the scale for the y-axis displayed on the right.

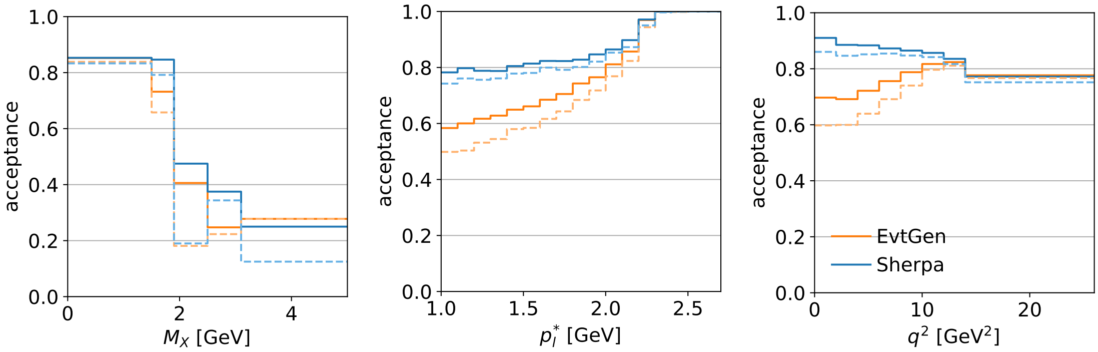

Figure 4.9: Signal acceptance as a function of M_X, p_ℓ^* and q^2 for NN_tight (solid lines) compared to NN_binned (dashed lines) defined in Eq. 4.4.2.

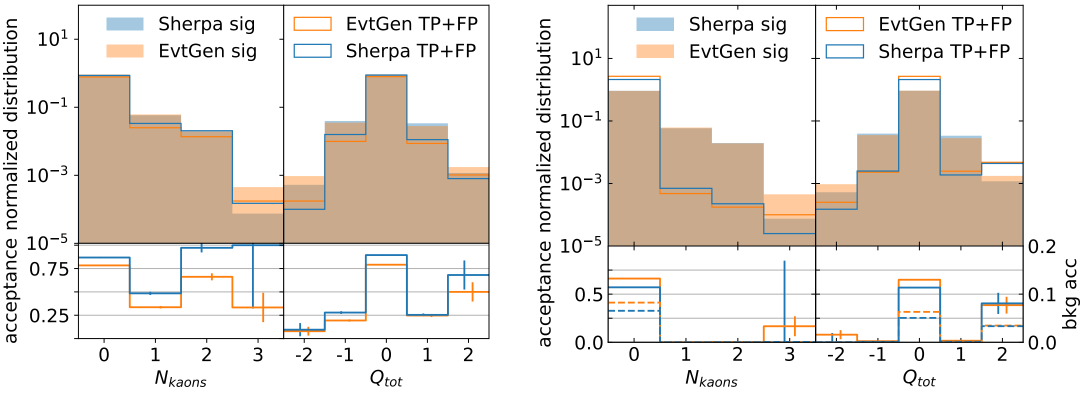

Figure 4.10: Q_tot and N_kaons distributions and signal acceptance for NN_tight (left) and NN_loose (right) trained on EvtGen data. For NN_loose the dashed lines in the lower panels show the background acceptance, using the scale for the y-axis displayed on the right.

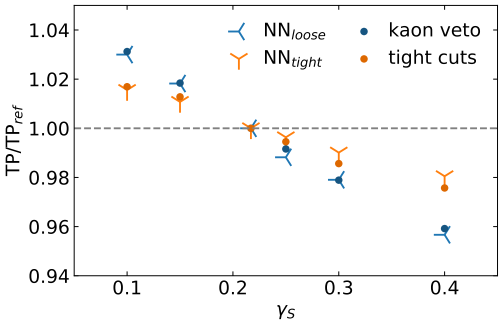

Figure 4.11: Sensitivity of the number of TP events to the $s\bar{s}$-popping probability γ_s. The number of TP events at the Pythia8 default is chosen as a reference value for each of the considered ML and cut-and-count approaches, $\text{TP}_\text{ref} = \text{TP}(\gamma_s = 0.217)$. The *tight* cuts are defined by the cuts listed in Eq. 4.4.4 plus $M_X < 1.5\,\text{GeV}$.

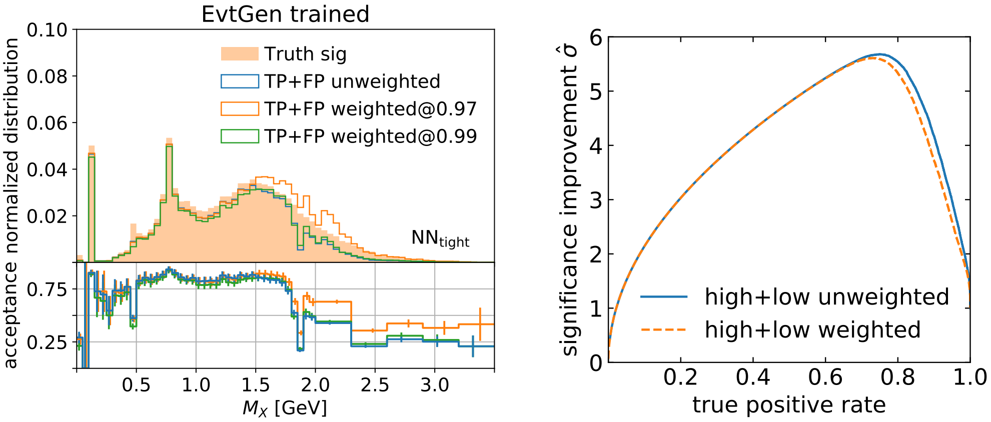

Figure 4.12: Acceptance in terms of M_X (left) and significance improvement (right) for the set-up including sample weights for EvtGen data.

Chapter 5

Deep Learning approach to strangeness tagging

When high-energy quarks and gluons collide as partons, they fragment and form successive branches of collimated partons. This process is known as parton showering. The partons in the shower will further branch out until they hadronise into hadrons due to colour confinement. This collimated bunch of partons is known as a jet and they can be seen as proxies to the high-energy quarks and gluons produced in a collision [4]. This chapter explores how jet substructure can help identify the quark flavour origin of jets. An overview of jet formation is given in Section 5.1 followed by a review on different types of jet tagging from simple deterministic approaches to application of machine learning algorithms. The focus is shifted from Section 5.3 towards strangeness tagging where we examine the potential of building a tagger among light jet backgrounds with neural networks (NN). This chapter is based on Ref. [4].

5.1 Jets

The quark model and the parton model were created to describe rather different physics: the former classifies possible states of hadronic matter, while the latter applies if we want to describe hadrons within high energy interactions. A parton can be understood as point-like constituent of the hadron carrying a fraction of the hadron momentum. The force between quarks is QCD with gluons as the force carriers as described in Section 2.1.1, an important feature of QCD is the strong coupling $\alpha_s = \frac{g_s^2}{4\pi}$, which in the framework of perturbative QCD becomes a scale dependent "running" constant. Within this framework, predictions for observables are expressed in terms of the renormalised coupling $\alpha_s(\mu_R^2)$ where μ_R is known as the renormalisation scale. This scale dependence leads to key properties where QCD interactions in the low energy regime are stronger than in a high energy regime. This effect is also known as colour confinement where quarks and gluons are strongly held together in the form of hadrons. In the high energy regime, QCD is asymptotically free which means the quarks are weakly interacting. Perturbative QCD utilises this property to compute strong processes given a high enough energy scale. This cut-off is known as Λ_{QCD} where further description of the process below this scale requires non-perturbative QCD. Subsequently, the parton model is theoretically justified as the lowest order approximation of a perturbative QCD calculation.

In a high energy collision event with protons, hundreds of particles are produced. Each proton contains numerous partons, each carrying a fraction of the proton's momentum. The partons of the two protons interact with each other via a large momentum transfer. The short distance interactions can be calculated perturbatively as mentioned above. The extraction of this calculable part from the non-perturbative part utilises parton distribution (or fragmentation) functions (PDF). These objects can be interpreted as probability distributions introduced by the parton model.

The wide energy gap between the proton mass and the fraction of the collision energy carried by the colliding partons is typically filled with emission of additional partons, which is referred to as initial state radiation (ISR). Note that the ISR is not always small because the hard[1] momentum transfer can be smaller and therefore, the ISR can still be considered hard. The hard interaction process determines the topological

[1] Hard, short distance, high energy are the same thing in this context.

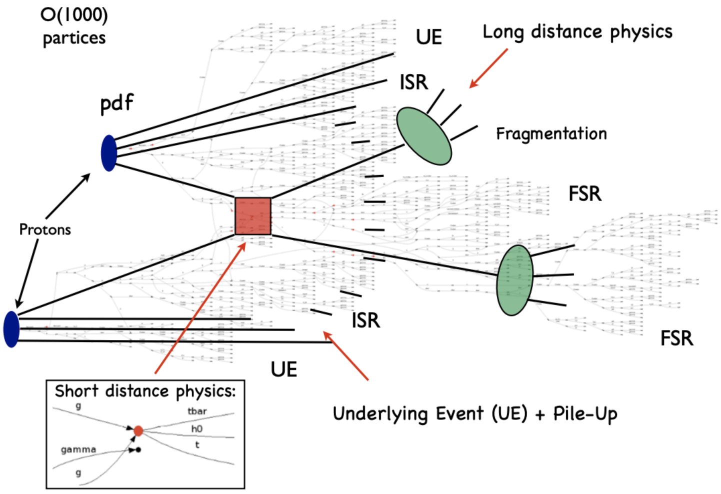

Figure 5.1: A schematic diagram representing a typical particle collision event [4].

structure and the composition of the final state. Furthermore, if colour-charged particles are produced in the hard interaction process, they are likely to emit even more partons, known as final state radiation (FSR), which bridges the gap between the interaction energy scale to Λ_{QCD}, where non-perturbative QCD arranges the partons into colour-neutral hadrons. The collection of collimated partons is known as a jet and they can be seen as proxies to the high-energy quarks and gluons produced in a collision.

While the hard process happens between the partons, the spectator partons still carry a proportion of the proton's energy. They are directed into the forward direction of the detector, but a non-negligible amount of radiation off these spectator partons can still end up in the central region of the detector. This type of excess measured is called the underlying event (UE). The schematic diagram of a collision is summarised in Fig. 5.1.

5.2 Jet tagging

When studying high-energy particle collisions, the hadronic final-states are the major source of information for researchers to study the underlying processes while establishing QCD as the fundamental theory of strong interactions [4]. Specifically, they allow for tests on perturbative QCD and tuning of Monte Carlo event generators. Hence, one often end up investigating the quarks and gluons produced as final states. However, the successive showering and hadronisation mean that the final state particles appear as collimated bunches of hadrons known as jets. There are various definitions for a jet depending on the jet algorithm employed. The most popular jet algorithm is known as the `anti-kt` algorithm [101]. It is part of the sequential recombination algorithm family which utilises concepts of minimal distance between recombined particles. For any list of particles, two set of distances are calculated. The first one is the inter-particle distance between any pair of particles (i, j) given as:

$$d_{ij} = \min(p_{T,i}^{-2}, p_{T,j}^{-2})\Delta R_{ij}^2 \ , \tag{5.2.1}$$

where $p_{T,i}$ is the transverse momentum of the ith particle and ΔR_{ij} is the geometric distance in the rapidity-azimuthal angle plane (η, ϕ) given as:

$$\Delta R_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2} \ . \tag{5.2.2}$$

The other set is the beam distance given as:

$$d_{i,B} = p_{T,i}^{-2} R^2 \ , \tag{5.2.3}$$

where R is the parameter known as jet radius. The algorithm iteratively finds the smallest distances among all d_{ij} and $d_{i,B}$, combine them into objects until the list is exhausted. If two objects are close in the rapidity-azimuth plane, they are likely to have come from the same parton and therefore are more likely to recombine. Similarly, when $\Delta R_{ij} > R$, the object is outside of the jet radius and is no longer recombined. The power of -2 on the transverse momenta allow this algorithm to further prioritises hard particles which avoids complications with soft radiation. This feature is favourable in an experimental context and hence explains its popularity.

Jet flavour identification is a crucial ingredient in measurements. Heavy flavour tagging is an active area of research as top and bottom physics have huge implications in Higgs related research and the flavour structure of the SM. Traditional tagging approaches were developed based on extensions of standard jet algorithms. The focus later shifted to the creation of innovative observables through inspecting jet substructures. Standard top quark taggers used by ATLAS and CMS utilised kinematic features induced by the top and W-boson masses [102–104]. Another well established top tagging method is a two step process based on the number of prongs within the jet, known as N-subjettiness [105], and the SoftDrop mass variable [106]. The situation is similar for bottom and charm tagging, existing algorithms typically use variables connected to the properties of heavy-flavour hadrons within the jet. The lifetime of hadrons carrying b quarks is of the order of 1.5 ps, while the lifetime for hadrons carrying c quarks is around 1 ps. This long lifetime (in particle physics sense) allows the hadrons to travel for a few millimetres depending on their momentum before decaying. This small displacement gives rise to displaced tracks in the calorimeter away from the primary vertex (PV) which allows us to reconstruct a secondary vertex (SV) [107]. The SV provides a distinctive feature for bottom and charm events, therefore algorithms have been built around reconstructing/hunting such displacements. Further information on the algorithms can be found in Ref. [108, 109].

Another well-studied area of jet tagging is quark-gluon discrimination. QCD backgrounds are simply dominated by gluon jets, strong discrimination is crucial in obtaining definitive tests on QCD, reducing background for various measurements and improving calibration for detectors. A well established list of traditional discriminants have been developed over the years [110]. Two main categories include the jet shapes and multiplicity-based observables.

For the multiplicity-based observables, a common choice is the iterated SoftDrop multiplicity (ISDm), the number of branchings which have passed the SoftDrop condition, as gluons tend to have a higher ISDm. An extensive review for some of these discriminants can be found in Ref. [4].

5.2.1 Machine learning in jet tagging

Notice that all of the tagging methods mentioned have mostly been simple univariate approaches, they have all worked sufficiently well to within acceptable systematic uncertainties for experimental usage. As data driven analytics evolve, the natural next step is to employ multivariate analyses such as machine learning or deep learning. Such techniques allow for direct analysis from low level detector information without constructing high-level engineered observables. Light quark tagging between up and down quark jets has shown promising result through an observable known as p_T weighted jet charge [111–113] in combination with low level track information [114]. This observable has been proven to be effective as a measurable discriminant [115–120] but machine learning further improved its performance significantly [114].

Furthermore, cross-disciplinary methods can be applied to particle physics data through deep learning. For example, natural language processing and computer vision techniques along with their neural network architectures. Heavy flavour tagging has since benefited from these advancements, in particular DeepCSV [121] and this deep learning based secondary vertex finder [122]. Both algorithms utilised cutting-edge sequence focused deep learning techniques and achieved strong performance in distinguishing heavy flavour jets from light and gluon jets. They can also be thought of as universal jet taggers within their limits.

Deep learning revolutionised quark-gluon classification into an image analysing task [22–24]. Energy depositions of particles within a jet is transformed into pixelated two-dimensional images in the (η, ϕ) plane with the pixel luminosity proportional to the energy carried by the particles. Different types of particles can be treated as different colours much like how coloured images are formed by stacking RGB grids. Convolutional neural networks can then process these "images" and classify jets initiated from quarks or gluons. This method works primarily because of differences in particle multiplicity and energy distribution between the two classes.

Notice that one type of quark is missing from all of the described taggers. The next section is a review of the latest development in strangeness tagging and an introduction to the study presented in the rest of this chapter.

5.3 Strangeness tagging

We have mentioned various algorithms and analyses on tagging different types of jets without any mention of strange tagging. There is currently no algorithm in use for identifying strange-quark jets at the LHC. Inclusion of such an algorithm would be complementary along with existing heavier quark tagging software for analyses like $t \to W^+ s$ [123] and $h \to s\bar{s}$ [124] decays.

Machine-learning methods have been explored to tackle strangeness tagging, Refs. [125, 126] utilised recurrent neural networks (RNN) as a feature extractor along with particle level 3-momentum, mass and high level jet related features to achieve strong separations. They compared potential performances under different detector settings which emphasised the importance of particle identification (PID). Alternatively, detector based track information combined with image recognition type neural networks were explored in Ref. [127]. This study included an additional step of tagging long-lived neutral kaons K_{L}^0 through different detector signatures. Both studies struggled to achieve classification comparable to the efficiencies seen in heavy flavour bottom or charm quark tagging. One of the reasons is that the strange hadrons, such as kaons, are abundant in all light quark fragmentations. In addition, strange hadrons are experimentally difficult to identify as short-lived K_S and Λ mesons decay within the inner tracking detectors and therefore can only be reconstructed through well-measured invariant mass of their decay products. It is possible to distinguish long-lived $K_\pm$ from other hadrons through Cherenkov detectors. Such PID capabilities are currently unavailable at ATLAS and CMS, as LHCb is the only detector within the LHC equipped with the RICH detectors [128]. However, time-of-flight detectors have been studied and planned for future upgrades [129, 130], more on time-of-flight is discussed in Appendix B.3. Further discussion on the importance of

PID is shown in Section 5.5.2.

The analysis shown in the rest of this chapter explores building a strangeness tagger against all light quark jets using neural networks for LHCb. Two directions are considered for this NN, the first studies the effect of how the jets are defined, one can label a jet with certain quark type given it is quark matched, but this procedure is not physical in an experimental environment where the leading jet is typically taken as the target jet. We study the consequences of training with quark matched jets while testing on leading jets. The other is feature selection in exploring jet substructure as a proxy to understand hadronic radiation patterns emerging from energetic (anti-)strange quarks. In particular, the usage of jet jet-flavour variable J_s introduced in [124].

The remainder of this chapter is organised as follow: Section 5.4 is the recipe for the simulated data including detector effects and cuts. Section 5.5.2 compares definitions and particle constituents between the jets. Section 5.5 describes feature sets and the performance of NNs trained with different features which further emphasis the importance of PID. Then, Section 5.6 showcases the result of the NN trained within the LHCb environment and a summary is given in Section 5.7.

5.4 Event generation and preprocessing

We studied the discrimination between light jets in $pp \to Z(\to ll)j$ process at the LHCb. All of the samples are prepared within the $2 < \eta < 5$ region in order to meet with LHCb specifications. The samples have been generated using Sherpa [93]. The parton level events are generated using AMEGIC++ [131] as ql^+l^- and $\bar{q}l^+l^-$ separately at $\sqrt{s} = 13$ TeV. The quarks are required to have a minimum of 20 GeV transverse momentum, where for leptons, this has been required to be 10 GeV. The minimum angular separation between two leptons and a lepton and a jet has been set to 0.4. Finally, the minimum same flavour lepton invariant mass has been set to 50 GeV. The parton shower and hadronisation are handled through COMIX [132] and CSSHOWER++ [133]. In addition, the samples are generated with NNPDF 2.3 PDF set [134] within LHAPDF package [135].

The preprocessing loosely follows the LHCb specifications [136] using MadAnalysis 5 version 1.9 [137, 138] alongside with SFS [139] machinery for detector simulation. The detector simulation mainly consists of transverse momentum (p_T) smearing and particle (mis)identification. Further details regarding the detector simulation can be found in the Appendix B.1; in the following, we will discuss the event selection and the requirements introduced for the preprocessing.

The jet objects are reconstructed using the `anti-kT` algorithm [101] embedded in FastJet version 3.3.3 [140]. The radius parameter for the jets has been chosen to be 0.5 with minimum transverse momentum at 20 GeV. In order to tag $b(c)$ jets, MadAnalysis 5's internal hadron matching has been employed with $\Delta R(j, B(C)) < 0.3$ and each tagged jet has been removed from the jet collection. Among these jets, light jets are selected within $2 < \eta < 4$. Similarly, lepton objects are selected if they satisfy $2 < \eta < 4.5$ and $p_T > 10$ GeV limitations. We employed simple ΔR-based isolation to separate jet and lepton objects from each other. Any jet objects within $\Delta R < 0.2$ vicinity of an electron are thus removed from the jet collection. Similarly, leptons are removed from the lepton collection if they lie within $\Delta R < 0.5$ of a jet object.

In selecting the two opposite-sign same-flavour leptons from the Z decay, the event is required to have two same flavour leptons and their invariant mass is within $m_Z \pm 30$ GeV. Additionally, each event is required to have at least one light jet. Once the requirements are satisfied, each jet goes through a parton matching procedure where a jet is tagged if the parton level quark is within $\Delta R < 0.5$.

Table 5.1 shows the difference in the cross section of the samples for different quark types where events with tagged jets and leading jets are shown separately. The difference between the values is due to the detector efficiencies where it is possible to have events with no quark-matched jets. We observed that the leading jets are tagged as quark matched jets for 98% of the events, and the second leading jets are tagged for

Process	quark matched [pb]	non-matched [pb]
$pp \to sll$	0.521	0.644
$pp \to dll$	5.018	6.548
$pp \to ull$	9.123	11.998
$pp \to \bar{s}ll$	0.515	0.633
$pp \to \bar{d}ll$	0.871	1.086
$pp \to \bar{u}ll$	0.591	0.731
$pp \to gll$	-	7.608
$pp \to bll$	-	0.175
$pp \to cll$	-	0.439

Table 5.1: LO cross sections from SHERPA for each jll process where the left column contains cross sections of the quark matched tagged jets and the right column shows the same when only the leading jets are selected. Samples with gluons, b and c quarks are included separately as reference.

only 1% of events. Admittedly $Z+q\bar{q}$ production can create a significant background for this channel where the Born level contribution is enhanced by gluon mediation. However, such processes are suppressed by α_S, and we calculated the corresponding cross-section, after the requirements mentioned above, for $pp \to s\bar{s}Z(\to ll)$ to be 1 fb. Hence these contributions are disregarded without loss of generality.

In order to use the particle identification information from the tracker, each track, generated within 10 mm - 1.16 m radial distance from the production vertex, have been matched with the jets with $\Delta R \leq 0.5$. Instead of the constituents within the jet, the information from these tracks has been used for the training and testing. In Fig. 5.2 the main differences between tagged, leading and second-leading jets are presented for s, d and u samples. The top panel shows the number of charged kaons (left) and pions (right) originated from the tracks matched with the jet in question, and the bottom panel shows the flavour discriminating observable J_s [124,141] given as:

$$J_s = \left(\sum_{\text{track}} \frac{1}{p^{||}_{\text{trk}}}\right)\left(\sum_{\text{track}} p^{||}_{\text{trk}} R\right) \qquad ; \qquad p^{||}_{\text{trk}} = \mathbf{p}_{\text{track}} \cdot \hat{\mathbf{p}}_{\text{jet}} \ ,$$

where $p^{||}_{\text{trk}}$ is the scalar projection of the track momenta on the unit momenta of the reference jet. R is an identification constant where it is -1 $(+1)$ for positively (negatively) charged kaon and zero for all other particles. The colour codes blue, red,

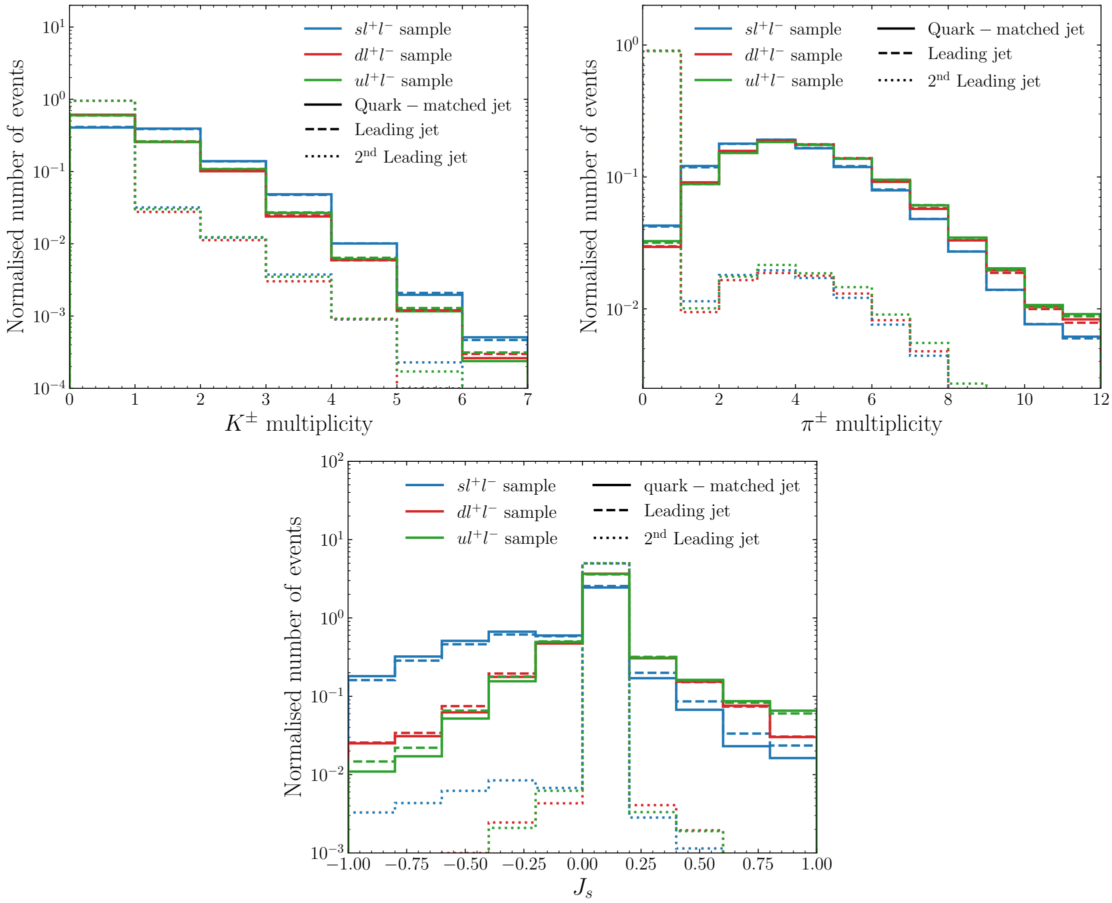

Figure 5.2: Distributions of the number of charged kaons (top left), charged pions (top right) and J_s where red, green and blue represents samples with d, u and s quarks and solid, dashed and dotted lines represent the histograms for quark matched, leading and second leading jets.

and green represent samples generated with s, d and u quark jets. In addition to the colours, the line style represents the nature of the jet where solid, dashed and dotted lines stand for quark matched, leading and second-leading jets respectively. Whilst kaon multiplicity indicates a slight difference between strange and other samples, the difference is relatively minor and does not propagate to the pion multiplicity histogram. However, we observe a significant difference in the J_s distribution which indicates its potential discriminating power. As mentioned in the earlier studies [127], the momentum weighted fraction of the charged kaons are significantly larger than the neutral kaons, which can live long enough to reach the calorimeter. Thus this analysis is aimed to exploit such hadron shower evolution over charged hadrons.

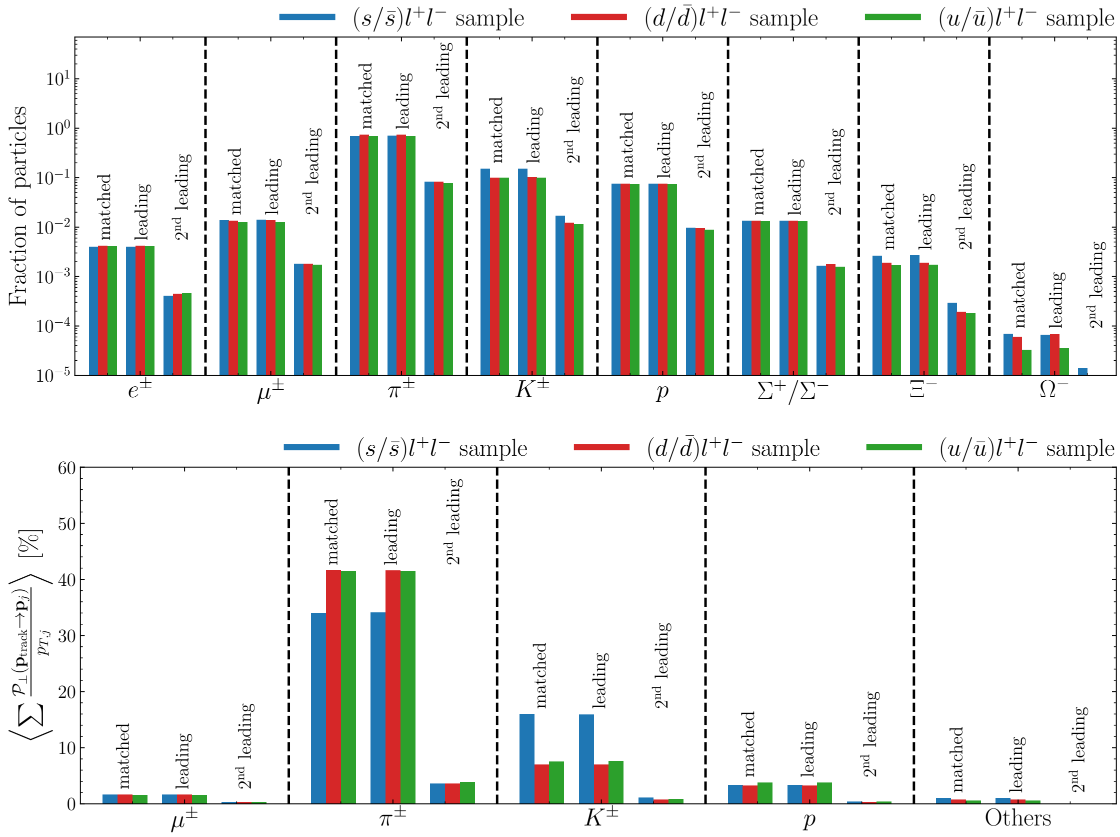

Figure 5.3: Fraction of track content of quark-matched, leading and second leading jets following the same colour-code as Fig. 5.2 including both q and $\bar{q}$ separately in the final state (Upper). Each particle includes fraction information for all three sets of jets as labelled above the bars. The bottom panel shows the mean fraction of transverse momentum carried by each type of particles for the quark matched, leading and second leading jets.

J_s captures the momentum fraction of charged kaons alongside the strangeness of the process, which allows for discrimination between a process with s-quark from $\bar{s}$-quark. Also, the serious shortage of strange hadrons in processes with other light quarks allows the s-type sample be segregated among others.

In the upper panel of Fig. 5.3, the fraction of identities from each track within jets is shown. Each charged particle has been divided into three blocks for quark-matched, leading and second-leading jets, and each jet block is further divided into three colour-coded samples, namely strange (blue), down (red) and up (green) type samples. Fraction of pions suppress all the other contributions in each block which is also captured in Fig. 5.2 with respect to kaon multiplicity. This is because almost half of the neutral kaons incapable of reaching the calorimeter decay into a pair of charged pions. Since 98% of the quark matched jets are also leading jets, respective blocks show a close correlation for each particle content. Each of these particles leaves a significant track behind where the decay lengths ($c\tau$) for $\pi^{\pm}, K^{\pm},\ \Sigma^{+},\ \Sigma^{-}, \Omega^{-}$ and Ξ^{-} are 7.8 m, 3.7 m, 2.4 cm, 4.4 cm, 2.4 cm and 4.9 cm respectively [142]. However, since not all of them are precisely identifiable, we will only use the identification information from charged kaons, charged pions, protons and muons, and the rest of the particle content within a jet is tagged as the same.

The bottom panel of Fig. 5.3 shows the mean fraction of transverse momentum carried by the respective type of particles within any given jet. For muons and protons, the difference between different samples and jet definitions are small. However, relatively large deviation for charged kaons and pions further supports the use of J_s and its component $p_{\rm trk}^{||}$ as features for our multivariate analysis. More on feature selection is discussed in the next section.

5.5 Identifying strange jets with deep neural networks

5.5.1 Features

The classification speed of a neural network, alongside its precision, are the core concerns for experimental collaborations. Hence, this study aims to devise a simple architecture that can achieve relatively high precision with the given feature space. To that end, we formed different groups of features and tried to understand their contributions to the given network. Tables 5.2 shows the features used, in particular J_s is the single high-level feature due to its discrimination power exhibited in the Fig. 5.2. The low-level features include the momentum fraction, particle identity

Name	Features & number of inputs	
High-level	J_s	1
Low-level	$\left\{\mathcal{P}'_{\text{track}},\ \text{PID},\ Q,\ d_0,\ d_z\right\} \in 5$ tracks ordered by $\mathcal{P}'_{\text{track}}$	25
Low + High	$J_s, \left\{\mathcal{P}'_{\text{track}},\ \text{PID},\ Q,\ d_0,\ d_z\right\} \in 5$ tracks ordered by $\mathcal{P}'_{\text{track}}$	26
Tracker	$\mathbf{p}_j, \{\mathbf{p},\ Q,\ d_0,\ d_z\} \in 5$ tracks ordered by $\mathcal{P}'_{\text{track}}$	39
Tracker + PID	$\mathbf{p}_j, \{\mathbf{p},\ \text{PID},\ Q,\ d_0,\ d_z\} \in 5$ tracks ordered by $\mathcal{P}'_{\text{track}}$	44

Table 5.2: The list of features used to study the contribution to the network. The braces indicate the list of features used for ordered tracks and the features out of the bracket are independent features. Boldface p stands for the particle momenta in polar coordinates; p_T, η, ϕ and energy where subscript j refers to the reference jet.

(PID)[1], particle charge, transverse impact parameter and longitudinal impact parameter for five leading tracks ordered by normalised projected momentum fraction of the tracks. In the following, we combined the high and low level features to observe if the PID alongside with projected momentum fraction of the track can suppress the importance of J_s. This will directly show if the PID information from pions and muons can impact the prediction outcome. In order to see if the neural network can reproduce the projected momentum fraction, we also studied a feature space including complete information of the three momenta of each track and the reference jet. This group of features is denoted as the tracker. Finally, we also included the PID along with the tracker group. Since predicting the outcome with a large set of inputs is more expensive, this test will reveal if the normalised momentum fraction is a satisfactory replacement for three-momenta. The following section focuses on the impact of these features in a neural network.

5.5.2 Performance comparison

In order to study the performance from each features group, we prepared two dedicated networks depending on the number of input features based on TensorFlow version 2.1 [143, 144] and Keras [145]. We have an architecture for the high level

[1] PID has been one-hot encoded for pions, kaons, protons and muons. Rest of the particle content has been identified as charged track.

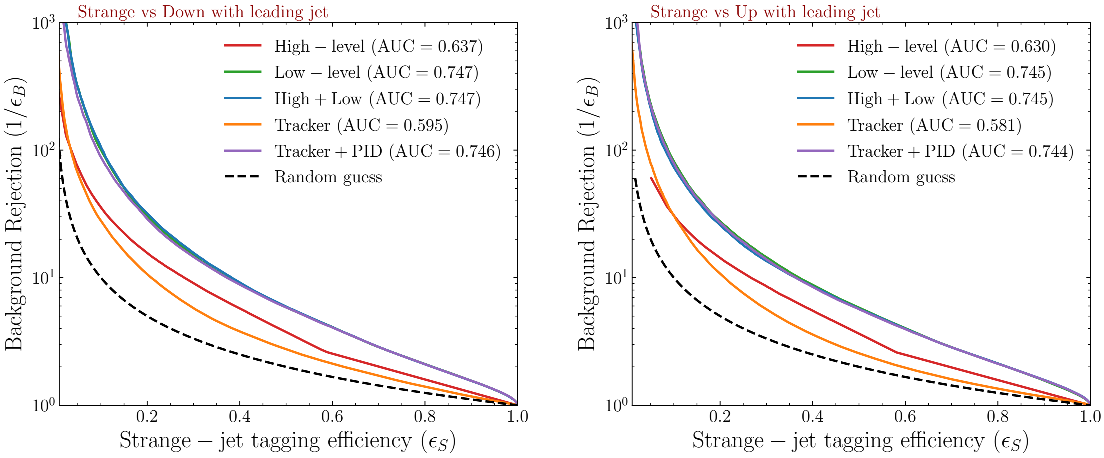

Figure 5.4: ROC curve for strange jet samples against down-type jet samples (left) and up-type jet samples (right). Colour code represents the feature mapping shown in Table. 5.2 where red, green, blue, orange and purple corresponds to high-level, low-level, high + low, tracker and tracker + PID, respectively.

only models with just one feature in Table B.2, the other groups all use the network described in Table B.1. All hyper-parameters shown for the architectures are optimised using `hyperopt` version 0.2.5 [146]. The data is standardised using the standard scaler from Scikit-Learn version 0.22.1 [147].

We train the NNs to classify the two main backgrounds. In other words, we focus on s-jets vs. d-jets, or s-jets vs. u-jets. Note that s-jets include both s and $\bar{s}$ events and similarly for d- and u- jets where the anti-quark events are included.

In order to achieve maximum efficiency, each network is trained using quark-matched jets. However, this methodology requires matching of parton-level quarks with the reconstructed objects, it is not accessible during an experiment. Hence leading jets were used for testing. As mentioned in the previous section, the jet definitions only differ by 2% in population. Therefore, we did not observe any significant difference between quark-matched and leading jet testing results.

Fig. 5.4 shows the Receiver operating characteristic (ROC) curve for each features set presented in Table 5.2. We have $pp \to (s/\bar{s})ll$ sample against $pp \to (d/\bar{d})ll$ sample on the left and the same signal sample against $pp \to (u/\bar{u})ll$ sample on the right. One immediately observes from both sets of ROC curves that the PID information

	Strange vs. Down		Strange vs. Up	
Features	$\epsilon_S = 80\%$	$\epsilon_S = 60\%$	$\epsilon_S = 80\%$	$\epsilon_S = 60\%$
High-level	1.430	2.507	1.415	2.421
Low-level	2.107	4.100	2.125	4.009
Low + High	2.095	4.098	2.128	3.978
Tracker	1.397	2.119	1.350	1.967
Tracker + PID	2.097	4.073	2.116	3.954

Table 5.3: Background rejection values for each feature group at 80% and 60% efficiency.

is crucial for the classification. The qualitative performance of individual groups are similar between the two classifiers, the tracker group scored the lowest, which is closely followed by the model trained with only the high-level feature J_s. The latter is expected to achieve minimal success; however, the fact that it manages to score higher than the tracker group emphasises its discriminating power and the importance of particle identification, especially for charged kaons.

We can observe that the rest of the feature groups all scored the same AUC value. The existence of particle identification seems to affect each feature group similarly. However, we observed varying training times where High+Low managed to converge faster (46 epochs for both down and up type characterisation) than only Low-level features (60 epochs for down and 47 epochs for up characterisation). Table 5.3 provides additional comparison between the two ROC curve where background rejection values are presented for each feature map at 60% and 80% tagging efficiency point. This further shows that the Low-level features give the most cost-effective results in our tests.

5.5.3 Understanding the features through SHAP values

We have shown the performance from each feature set and the importance of PID in the previous section, this section is a deeper investigation on the individual features and how they affect the learning of the NNs. We employ Shapley values through the SHapley Additive exPlanations (SHAP) [148] package as a probe to understand how different features affected the performances shown in Fig. 5.4. Shapley values measure

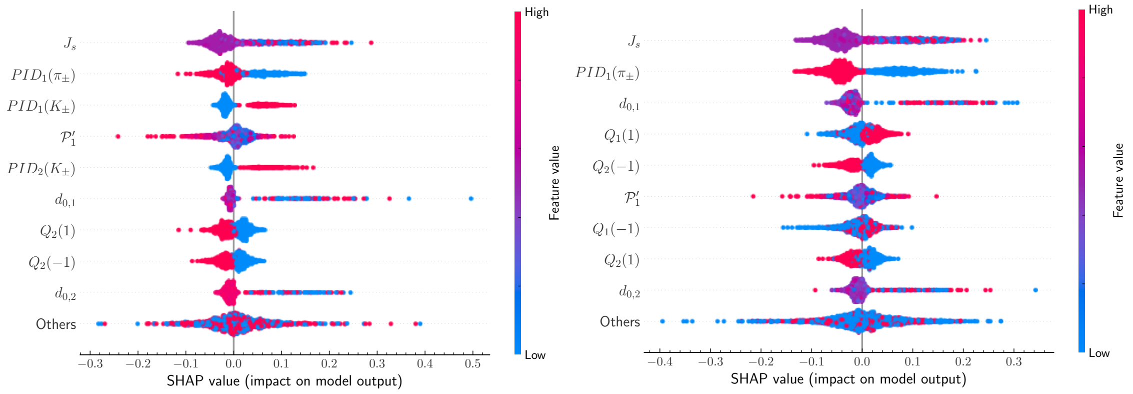

Figure 5.5: Top 10 features with the most impact on the classification output according to SHAP values for high + low features. The left panel follows the strange vs down classification where right panel shows strange vs up classification.

the average marginal contribution of each feature in augmenting the classification performance as part of a group of features.

Classical Shapley values were first utilised in game theory to compute explanation of model predictions [149–151]. Consider a value function Φ for all features in a set S, where S is the feature set excluding x_j written as $S \in \{x_1, ..., x_p\} \backslash \{x_j\}$. The feature x_j is isolated from S such that the marginal value predicted by the model with and without this feature can be computed. The equation for it can be written as [150]:

$$\Phi_j(\nu) = \sum_{S \in \{x_1,...,x_p\} \backslash \{x_j\}} \frac{|S|!(p - |S| - 1)!}{p!} (\nu(S \cup x_j) - \nu(S)) \ , \tag{5.5.1}$$

where p is the total number of features including the isolated ones, x is the vector of features and ν is the model prediction. In other words, the Shapley value computes the contribution of each feature to the model prediction, weighted and summed over all possible contributions. A positive SHAP shows the influence from the feature for the model to classify the particular test set as a signal. Modern usage of SHAP follows the same principles, the difference is mainly on how the model predictions are estimated for different types of algorithms. A comprehensive review of SHAP can be found in Ref. [148].

Figs. 5.5 and 5.6 show the 10 most influential features in order of SHAP with the top being the most important for high + low and the tracker + PID groups. In

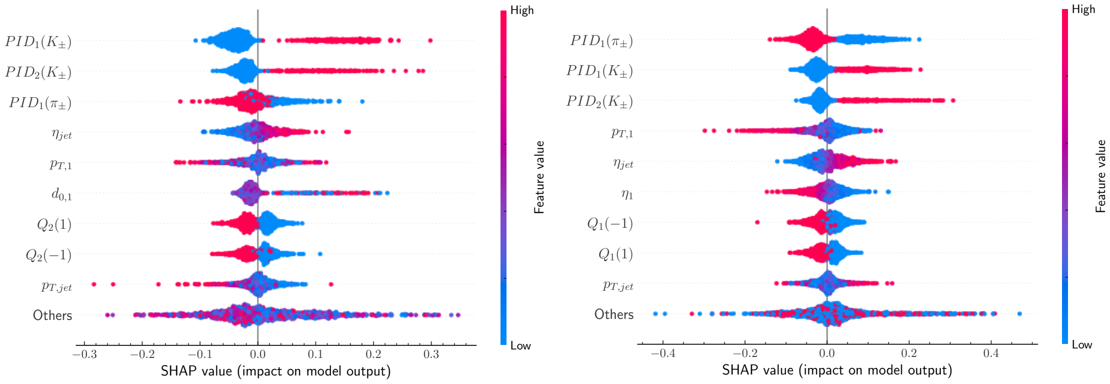

Figure 5.6: Top 10 features with the most impact on the classification output according to SHAP values for Track+PID features. The left panel follows the strange vs down classification where right panel shows strange vs up classification.

each of the figures, the left panel shows strange vs down quark results, where the right panel shows the strange vs up quark results. The subscript in each feature represents the position of the object on $p^{||}$-ranked sequence, and the parameter in parenthesis shows the specific element after one-hot encoding. The colours represent the size of the feature, for instance, $PID_1\pi_\pm$ is whether the track with the highest $p^{||}$ is a charged pion. The blue population are mostly positive in SHAP which means that events without charged pions as their highest $p^{||}$ tracks are likely to be signal events.

From Fig. 5.5, J_s ranked as the most influential feature among all. We know it has allowed the network to converge faster from the number of epochs trained when comparing with only low-level features. In both cases, the models tend to treat event with J_s near its extrema as signals. This is expected as there are strong overlaps between different quark types at $J_s \approx 0$ as illustrated in Fig. 5.2. Hence why more of the purple population has negative SHAP values. In addition, features related to charged kaon identification are crucial in strange jet classification with both models highly ranking these features.

Similar behaviour is observed in Fig. 5.6 where charged kaons play an important role in signal classification. It is interesting to see that charged pions are relatively effective in classifying background events when they are so abundant as a final

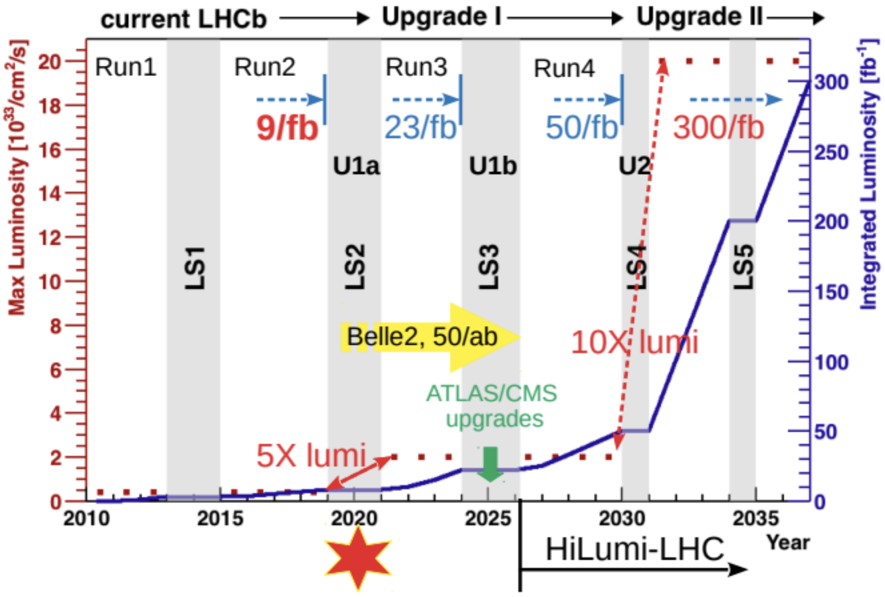

Figure 5.7: Predicted integrated luminosity of LHCb from 2010 - 2037 [152, 153].

state hadron for all quark types. We also observe that transverse momentum and pseudo-rapidity information are effectively used alongside particle identification; this suggests that the network is also trying to reconstruct a J_s-like observable to achieve better classification. Additionally, due to the relatively long lifetime of charge kaons and pions, the transverse impact parameter is valuable for signal classification.

5.6 Prospects at LHCb

The amount of data collected at LHCb through Run1 and 2 is at around 9 fb^{-1}. The total has been planned to increase to around 50 fb^{-1} from now till the end of Run4 and 300 fb^{-1} after Upgrade II as shown in Fig. 5.7 [152, 153].

This section is an exploration in applying the method described above against all light quark jet backgrounds. We build a classifier for s-jets vs d-jets, u-jets and gluon-jets as a feasibility case study. The classifier employs the same architecture as described in Table B.1 and the features used are the low level group from Table 5.2. Key assumptions include perfect discrimination of $b-$ and $c-$jets events and the gluon contribution is generated with the same kinematic specification described in

	Strange vs. all other light jet background		
Features	$\epsilon_S = 80\%$	$\epsilon_S = 60\%$	AUC
Low-level	1.995	3.745	0.73

Table 5.4: Background rejection values at 80% and 60% efficiency points and the AUC for the LHCb scenario trained low level features.

Section 5.4. Note that only the leading jets are selected for the gluon sample as it is not reasonable to quark-match gluon jets.

The fitted result is shown in Table 5.4. It is clear that the background rejection values are similar to the s-jets vs d-jets result shown in Fig. 5.4 and Table 5.3 even though there are so many more background processes involved. This suggests that the both the signal and the combined background machine learning efficiencies perform similarly to the simpler case shown above. A similar analysis for general detectors has been included in Appendix B.3.

5.7 Summary

In summary, this chapter explored the application of simple neural network architectures along with detector track information to identify strange-quark jets from other light-quark jets including up and down. We set up five different feature sets as shown in Table 5.2 around particle 3-momentum, the flavour observable J_s and its compositions, the particle/track momentum fraction with respect to its jet $\mathcal{P}'_{\text{track}}$. We studied the features on classification between s-jets with the two largest light-jet backgrounds, d-jets and u-jets separately. The models trained with only low level, high and low level, and the track + PID features performed similarly and that these three performed significantly better than the remaining two sets without PIDs. This clearly demonstrated the importance of PID as the models recreate something similar to J_s within their latent spaces. This is further supported by the SHAP values analysis as $\mathcal{P}'_{\text{track}}$ and PIDs are highly ranked in feature importance. The result obtained from the systematic feature study indicates that $\mathcal{P}'_{\text{track}}$ is a strong candidate in replacing track 3-momentum which could help reduce dimensionality when more particles/tracks are considered.

As a feasibility study, we applied this method to distinguish s-jets from all light -jet backgrounds and gluons. Only low level features were used and the fitted result showed little difference from the above models in terms of ML signal and background efficiencies.

This study provided some insights into feature selection for strangeness tagging but the network architecture is too simple to make any further claims on the effectiveness of the presented architecture in a practical setting. A deeper analysis with more complex architectures and the complete set of transfer functions for detector effects is required. In addition, the number of tracks per jet is fixed and hence there could be more information when the whole jet is included through a geometry aware graph network.

Chapter 6

Predicting Realised Volatility with Deep Learning

6.1 Introduction

Deep learning (DL) is a blooming field of research and its capabilities for real world problems are being explored by different sectors. The financial industry is certainly one of the leaders in adapting machine learning (ML) and DL into their solutions for real world problems. The growth is immense from analytical/visualisation tools to modelling the stock market. This chapter is inspired by the research project in collaboration with Optiver aimed at investigating the potential in utilising deep learning techniques to predict realised volatility of stock market indexes.

Since the 2008 financial crisis, risk management has been ever more important for investors. The financial instrument commonly used to balance their risks and rewards are options. Options give their holder the right to buy or sell some underlying for an agreed-on price at a fixed expiry date. The underlying here can be stocks, indexes or other structural products. The ability to predict volatility accurately is crucial for pricing and trading these options.

There are two types of volatilities: implied (IV) and realised (RV). Implied volatility is a statement about the value of the options being traded based on opinions of market participants and realised volatility comes from the underlying's actual volatility observed retrospectively [154]. Further explanation on these two quantities and how they are derived and used will be discussed in the next section.

The traditional approaches for predicting RV often involve fitting an autoregressive (AR) model such as GARCH [155, 156] or a Heston model which assumes volatility is driven by its own stochastic process [157]. AR models are the most widely used family of models in literature due to their ability to empirically fit volatility clustering from financial time series. However, they are also known to lack flexibility in their fits especially for large prediction horizons [158]. Development of GARCH models is an ongoing field of research where various modifications have been produced over the years leading to more flexible models, for instance, Engle and Lee suggested a two equation model where each of them represents long-run and short-run components of volatility [159]. On the other hand, Heston models assume stock prices undergo Brownian motion. The model Heston derived follows an Ornstein-Uhlenbeck process and the parameters for such models are determined through either the Generalized Method of Moments [160] or simulations [161].

Majority of production level solutions have turned to machine learning approaches such as Ridge Regression and Random Forest. They are favoured because of their transparency and reactivity to sudden market movement on top of simply better predictions over AR models. We mainly considered DL methods and there have been numerous papers on this subject. For example, Ramos-Pérez et al. [162] used predictions from a number of ML algorithms (Random Forest, Support Vector Machines and Gradient Boosting) in addition to historical information extracted from the underlying and formed an ensemble model with a neural network. Zhou et al. [163] utilized alternative data such as search engine volumes for their long-short term memory networks (LSTM). The models we present here include typical neural networks, Bayesian neural networks and various training schemes to maximise performance.

This project is in collaboration with Optiver. They are the leading company in option trading as a market maker. A market maker is someone whose principal trading method is to quote a two-sided market (a bid and an offer) [154]. We were part of the statistical arbitrage (statarb) research team which focus on systemetic trading.

The rest of this chapter is organised as follows: Section 6.2 concentrates on giving insights about options and why accurately predicting RV is important. Section 6.3 focuses on the data used, the choice of features and the preprocessing scheme. Section 6.4 is about the models chosen along with introduction to what they are before presenting the fitting result and backtesting result in Section 6.5. Section 6.6 is the final discussion.

6.2 Theoretical background

6.2.1 Option

Options are a type of derivative meaning their value are derived from the value of another asset. An option contract give its holder the right but not an obligation to buy or sell the underlying at an agreed-on price and date, this price is also called the strike. The cost for such a contract is called the premium. There are two main styles of options: European and American style. The difference is in their execution where the European style option can only be used at expiry but an American option can be exercised at any time before expiration. There are two general types of options, puts give the holder the right to sell the underlying for the strike at expiry and calls allow the owner to buy the underlying with the strike at expiry. In other words, puts bet on the underlying to lose value and calls bet on the underlying to gain value.

Options are typically used as a tool to manage the risk of the investment portfolio. For example, an investor is holding a long position over a stock, this investor would then buy a put to protect that position. In simple words, this means an investor bought some shares of a stock expecting their value to go up (long position), this investor then bought a put contract which gives the investor the right to sell those shares for an agreed-on lower price at expiry in case the stock drops in value. The put option is essentially an insurance for taking that long position, the maximum loss for the investor is then the premium from buying the put contract and the difference between the initial stock price paid to buy those shares and the predetermined strike price stated in the put. The action of buying an insurance is also known as hedging where the investor hedges the risk from taking the long position away with the put. More on hedging is discussed later this section.

Options can also be used speculatively i.e. an investor expect a stock to gain value, this investor can purchase a call and potentially make a large profit while only paying for the premium instead of the larger sum required to directly buy the underlying and left exposed to other risks. The maximum loss from this trade is just the premium paid for the call contract. There is another level to this where the investor can buy an out-of-money call option, such options typically have cheap premiums and are worthless if exercised against the current underlying price.

The value of an option can be separated into two parts as follows:

$$\text{Option value} = \text{intrinsic value} + \text{time value}\ . \tag{6.2.1}$$

For call options, the intrinsic value is simply the underlying's price at expiry subtracted by the strike. Note that if the calculated value is negative, it simply means the option is worthless. The moneyness of an option is determined by its intrinsic value where a contract with positive intrinsic value is said to be in-the-money. If the strike is the same as the current price of the underlying, such options are said to be at-the-money. A put option with 0 intrinsic value due to the strike being higher than the underlying is called out-of-money. The equivalent situation is true for call options. Time value of an option is at its greatest when the option is at-the-money, otherwise it is a decaying quantity as the expiry approaches. Moneyness is also closely related with the Greeks which describe the risk profile of options, more on the Greeks will be discussed below in this section.

6.2.2 Pricing options

There are various ways to price options, the most famous method is the Black-Scholes-Merton formula (BSM) which won the Nobel Economics Prize in 1997 [164–166]. It can be written as:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} - rC = 0 \ , \tag{6.2.2}$$

where C is the value of the option as a function of the stock price S and time t, σ is the standard deviation of the underlying's returns also known as the volatility and r is the interest free rate. The BSM framework requires some keys assumptions and they are:

- The underlying is tradable.
- A single constant interest rate r.
- No additional income from the underlying e.g. dividends.
- Ability to short the underlying.
- Constant volatility.
- The underlying changes continuously and therefore the investor can also continuously hedge.
- Volatility is the only parameter required to specify the distribution of the underlying's returns.

A tradable underlying means it is purchasable and sellable quickly and fairly for the size required to dynamically hedge, in other words the underlying needs to be liquid. An investor is shorting an underlying when shares are borrowed and sold at expiry betting on the underlying to lose value. Notice that these assumptions are mostly not applicable to the real market, the prominent example being volatility where it is obviously not constant. In addition, continuously hedging is not feasible because there is a cost for every hedge. Finally, assuming volatility is the only parameter for the returns distribution in turns assume normal or log-normal price distribution which is simply incorrect as we will see later this chapter. The BSM framework is clear and robust, it can manufacture an option and tell you the cost. However, this is only true under these idealised assumptions. The market does not behave like the assumptions but it can nevertheless provide a qualitative view for traders to adjust for these drawbacks.

Within the BSM framework, there is no analytical solution for American style options but there are for the European style. The solutions for European calls and puts are given as:

$$\begin{aligned} \text{call} &= S\exp((b-r)t)N(d_1) - K\exp(-rt)N(d_2) \\ \text{put} &= -S\exp((b-r)t)N(-d_1) + K\exp(-rt)N(-d_2) \end{aligned} \tag{6.2.3}$$

where

$$\begin{aligned} d_1 &= \frac{\ln(\frac{S}{K}) + (b + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}} \ , \\ d_2 &= \frac{\ln(\frac{S}{K}) + (b - \frac{\sigma^2}{2})t}{\sigma\sqrt{t}} = d_1 - \sigma\sqrt{t} \ , \end{aligned} \tag{6.2.4}$$

K is the strike, b is the generalized cost of carry parameter, $b = r$ gives the standard Black-Scholes stock option model, $b = 0$ gives the Black futures option model which is an adjusted version of the BSM to model futures contracts. Futures are similar to options but the holder has to exercise the contract at expiry. Lastly, $b = r - q$ where q is a dividend yield allows for adjustments on effective interest rates by approximating the dividend stream with that yield. $N(x)$ is the cumulative normal distribution function as shown below:

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(\frac{-z^2}{2}\right) dz \ . \tag{6.2.5}$$

The partial derivatives in Eq. 6.2.2 can be represented by Greeks written as:

$$\begin{aligned} \theta &= \frac{\partial C}{\partial t} \quad , \\ \Delta &= \frac{\partial C}{\partial S} \quad , \\ \Gamma &= \frac{\partial^2 C}{\partial S^2} \quad . \end{aligned} \tag{6.2.6}$$

There are 4 major Greeks representing different types of risks. θ is the rate of change of the option price with respect to time which can be thought of as the amount by which an option's value will decline every day. Theta PnL describes the profits and losses associated with θ, it is negative for option buyers as time works against long positions. On the other hand, option sellers benefit from option value decay where they could then buy these contracts back with a cheaper price and profit from the difference in the premium.

Δ is the rate of change of the option price with respect to the underlying which means it indicates how sensitive the option price is to movement in the underlying. Δ is also a measure of moneyness as mentioned above. If the strike is the same as the underlying i.e. the option is at-the-money. An at-the-money call contract has $\Delta = 0.5$ and -0.5 if it is a put instead. A call option is in-the-money when $\Delta > 0.5$ and out-of-money when $\Delta < 0.5$. The negative equivalent apply for puts.

Γ is the rate of change of Δ with respect to the underlying. If Δ is the speed of the underlying, Γ is the acceleration. It is very small when the option is deep in or out of money close to the expiry because Δ will either be close to 0 or 1 in these two scenarios and stays the same. It is at its largest when the option is at-the-money as expiry approaches because Δ rapidly jumps around 0.5. Γ and Θ have a cancelling effect on each other, this will be clearer later when the general PnL is defined.

The last Greek is Vega, it is the sensitivity measure of the option price with respect to change in implied volatility of the underlying. It is written as:

$$\text{Vega} = \frac{\partial C}{\partial \sigma} \quad . \tag{6.2.7}$$

Note that Vega is not a real Greek letter but a word created by option traders. It changes when there are large price movements in the underlying leading to increased volatility, and falls as the option approaches expiration. There are many other Greeks associated with other kinds of risk but these 4 have the biggest impact.

6.2.3 Delta hedging

Delta hedging is an option trading strategy aimed at reducing, or hedging, the directional risk associated with the price movements in the underlying. This approach uses options to offset the risk of either a single other option holding or an entire portfolio of holdings. The investor tries to reach a delta neutral state ($\Delta = 0$) and not have a directional bias [154].

Consider a stock option with price C bought at implied volatility σ_i, financing the purchase by borrowing at the risk-free rate r. This option is delta hedged by shorting the underlying S by Δ_i units where Δ_i is the delta hedge ratio calculated according to the implied volatility.

Time	Option position	Stock position	Cash position	Net
t	C^i	$-\Delta_i S$	$\Delta_i S - C^i$	0
$t + dt$	$C^i + dC^i$	$-\Delta_i(S + dS)$	$(\Delta_i S - C^i)e^{rdt}$	?

Table 6.2.3 breaks down the positions at different time points. At time t, the stock option is hedged by Δ_i units. The premium paid is C^i and the value gained from selling the underlying (the short position which is also the hedge in this case) is $\Delta_i S$, the subtotal on cash is then the two added together. We are holding onto the option contract and shorted stocks and therefore the net value at time t is 0. At time $t + dt$, the option position has moved by dC^i hence we need to further hedge the change in the movement of the underlying dS. Now we would like to know what is the net value at $t + dt$.

PnL is typically given as:

$$\text{PnL} = \text{Theta PnL} + \text{Gamma PnL} + \text{other effects} \tag{6.2.8}$$

The incremental PnL according to the table is then:

$$\begin{aligned} d\text{PnL} &= (C^i + dC^i) - \Delta_i(S + dS) + (\Delta_i S - C^i)e^{rdt} \\ &\approx dC^i - rdtC^i - \Delta_i dS + \Delta_i Srdt \end{aligned} \tag{6.2.9}$$

Note that $\exp(rdt) \approx (1 + rdt)$ as $rdt \ll 1$. From Eq. 6.2.9, we need two additional substitutions given below:

$$dC \approx \theta dt + \Delta_r dS + \frac{1}{2}\Gamma(dS)^2 \approx \theta dt + \Delta_r dS + \Gamma\frac{S^2\sigma_r^2}{2}dt \tag{6.2.10}$$

$$\theta dt = -\frac{1}{2}\Gamma S^2\sigma_r^2 dt - rdt\Delta_r S + rdtC \tag{6.2.11}$$

where Eq. 6.2.10 originates from the Taylor expansion of C up to $O((dS)^2)$ and Eq. 6.2.11 is just a rearranged version of Eq. 6.2.2 multiplied by dt on both sides. Substituting both equations into Eq. 6.2.9, we would obtain:

$$d\text{PnL} = \frac{1}{2}\Gamma S^2(\sigma_r^2 - \sigma_i^2)dt \tag{6.2.12}$$

which implies

$$\text{PnL} = \frac{1}{2}\int_0^T \Gamma S^2(\sigma_r^2 - \sigma_i^2)\exp(-rt)dt \tag{6.2.13}$$

Note that this equation has huge implications because there aren't any random components at all meaning the PnL is deterministic. This method of trading is also known as gamma scalping where the investor can now long options when RV>IV and collect profits given an accurate prediction of σ_r. The same is true when RV<IV and the investor shorts the options. Gamma scalping is incredibly powerful as money can be made from just market movement/volatilty, the reason why people haven't all gone crazy about option trading is because Γ comes with a cost θ, time is a risk as unexpected events could still happen. Notice that we have demonstrated the profitability of having accurately predicted realised volatility but none of the calculations actually require any knowledge on what RV is. The next section will focus on the main character of this project, RV.

6.2.4 Volatility

There are two types of volatilities as mentioned before: implied and realised. Implied volatility is the volatility set by market participants when they are pricing options. In other words, one can extract the implied volatility used to price the option through a pricing a model like the BSM framework given the other required variables. Realised volatility is simpler, it is essentially the historical standard deviation of the returns for a given underlying given as:

$$\mathrm{RV} = \sqrt{\frac{252}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} R_i^2} \ , \tag{6.2.14}$$

where 252 is the number trading days in a year included to annualise the measure, w is known as the winddown which is a quantity used to weight different time points throughout the day, R is the log returns and n is the number of recordings we are rolling over e.g. if R is recorded daily and we want to calculate the 5 days rolling RV then $n = 5$.

It is crucial to know that RV behaves differently in different regimes, these regimes could be a bearish/bullish underlying market, which means the value of the underlying is decreasing or increasing respectively, or it could be an unseen regime where no other periods in the past have similar behaviour. The main challenges in predicting RV are that we do not know when regime shifts occur and even if we do, it is not guaranteed the behaviour stays the same in the same type of regimes now compare to 20 years ago. Both RV and IV exhibit some level of mean reversion where they every so often would revert to their long-term mean. This feature is popular among trading strategies but lack reactivity to sudden movement in the underlying. This strategy remains effective as RV would typically increase when down ticks occur in the underlying and decrease slowly when the underlying goes up, more on this will be discussed in the next section when the data is introduced.

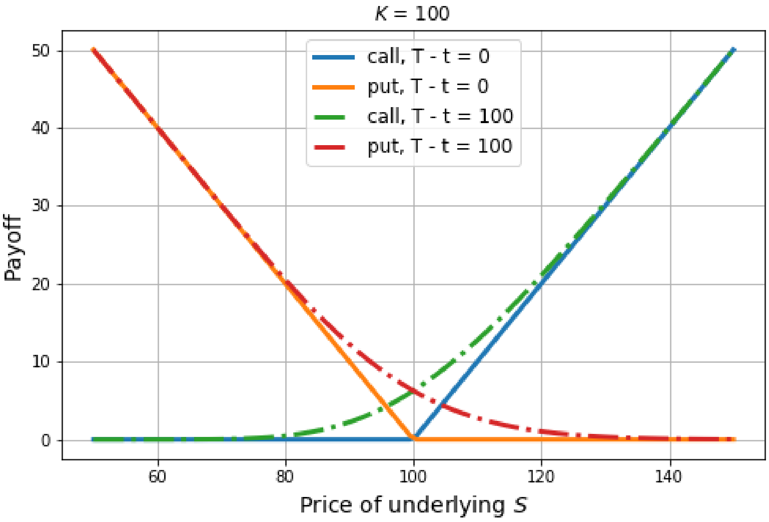

Figure 6.1: Payoff of a straddle at different time to expiry where T is the expiry and t is the current time.

6.2.5 Straddle

There are many different option trading strategies, most of them were created to mitigate a certain type of risk. We will focus on straddles as they are the simplest strategy to trade with volatility. Long straddle is essentially a long call combined with a long put, typically at-the-money ($\Delta = 0.5$), bought with the same strike and expiry. This type of trades ignore the direction moved by the underlying and collect positive PnL given the movement is large enough to cover the premiums. An example of a straddle bought at strike=100 is shown in Fig. 6.1. The minima of the payoff smile goes up with more time away from expiry, increase in volatility has a similar effect.

6.3 Data and Features

We used Standard & Poor's 500 (S&P 500) data dated from January 2014- September 2020. It is an index based on the top 500 companies in the United States weighted by their market capitalisation. We typically use 2014-2017 as the training set, 2018-2019 as the test set and keep 2020 completely out of sample until the final test. More on

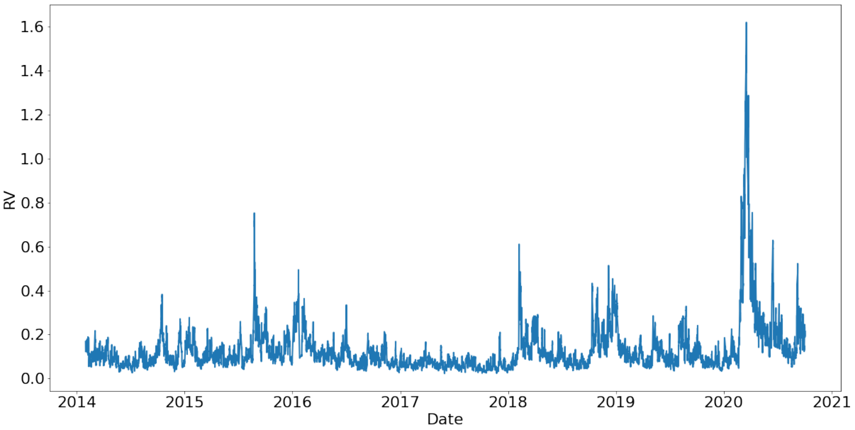

Figure 6.2: Rolling 1 day realized volatility from 2014 - 2020.

train test split is discussed in the next section. The data consist of 56 data points per trading day from 8:15AM to 10PM in 15 minutes intervals, this period includes both EU and US trading hours. A plot of this data is shown below in Fig. 6.2. The first obvious observation one can made is that there are some big spikes occasionally. Otherwise, the RV remains quite flat over a long period of time. Most of these big spikes are typically events, more on events are discussed in the following section.

6.3.1 Events

Events from 2016 onwards have been categorised based on their type. There are generally two types of events, we have recurring events such as the European Central Bank (ECB) press conference which makes the latest decisions on monetary policies. There can also be instantaneous events like when the US President Donald Trump tweeted about China trade tariffs causing strong market movements. Market participants often price in possible outcomes of events which cause the IV to gradually increase up till the end of the events. The IV tends to rapidly revert towards its long term mean after the event. The effect from recurring events are not very strong for the S&P because of the sheer number of companies averaging them out. However, big events like Covid-19 certainly have longer-term effects as investors remain cautious due to uncertainties on future development, a similar effect was observed after the

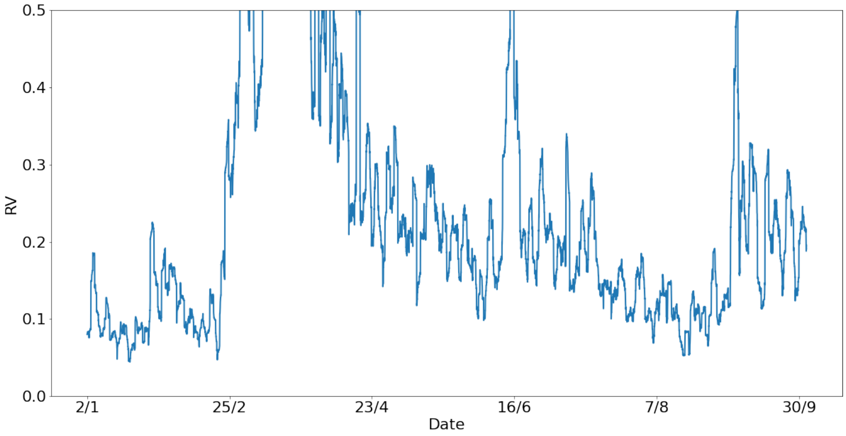

Figure 6.3: Rolling 1 day realised volatility zoomed into 2020.

2008 financial crisis. This kind of reaction keeps the long term volatilties at a higher level as shown in Fig. 6.3. The mean RV before Covid-19 reaction is about 0.12 and it became roughly 0.2 after the Covid-19 peak. This nicely echos the point made in the previous section where RV behaves differently in different regimes.

6.3.2 Features

Features from historical market movement include rolling averages over 0.5, 1, 2, 3, 5, 10 and 20 days of RV and moves. Notice that all of these features can be derived from the log returns with Eq. 6.2.14 for the RVs and moves are just rolling averages of itself. The RV features provide a sense of regime and the overall level the model should predict. The moves features are particularly important as they provide a sense of direction and amplitude for the prediction. Additionally, we have included datetime information as features. There are further explanation to this inclusion later in this section.

The target label is the 1 day out rolling 1 day RV meaning we took the rolling 1 day RV and shifted it by 56 data points. Due to the target originating from one of the features, it is extremely important to keep an extra day of data points between training and test set to avoid data contamination from future information. Otherwise, the final part of the training labels will be the same as the beginning of the test features.

Datetime effect

We have plotted the mean of the rolling 1 day realised volatility grouped by different time frames in Fig. 6.4. The most prominent datetime effect comes from weekdays as shown in the top left plot. It is clear that the RV is higher on Mondays and Fridays. We have included other effects such as hour of the day, week of the month and month in the rest of Fig. 6.4.

The hour effects are small and they originate from numerical error when recording. However, pattern could still emerge when combined with other datatime features. The monthly effect is seasonal where investors often re-balance their portfolio in the beginning of the year and activity slows down come summer till September and October when people are back in their offices. The week of month is included in hope of capturing some of the contribution from recurring events and other seasonal patterns. In practice, we use the datetime from the target as we should know exactly when we are predicting and these features are one-hot encoded to avoid simple numerical biases. In addition, month and week of month effects are weakened by multiplying with 0.1 after one-hot encoding to prioritise shorter term patterns.

6.3.3 Preprocessing

In most DL research, standardisation or normalisation of the data is common as they help the algorithm generalise better and therefore obtain better result. We have decided against both preprocessing techniques.

First of all standardisation transform the data forces a zero mean and the standard deviation becomes one, such transformation works best when the data is Gaussian. We cannot make this assumption on RV as shown in Fig. 6.5 where the distribution is heavily right-skewed along with a strong tail. Normalisation typically transforms the data to between 0 and 1 through manipulation with the minimum and maximum

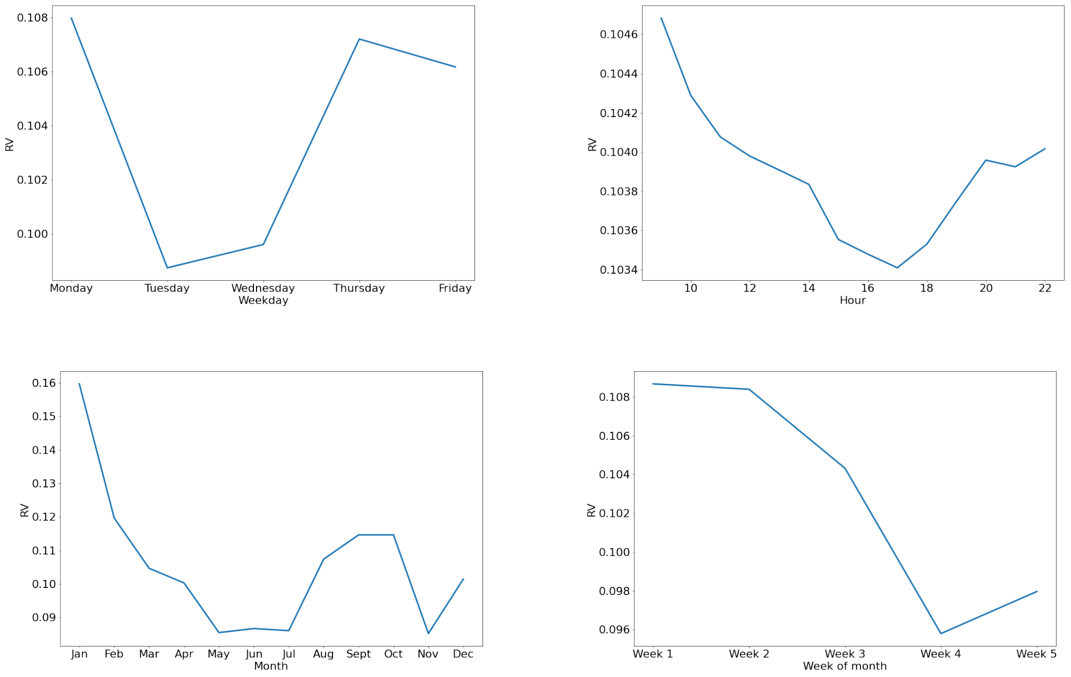

Figure 6.4: Mean of rolling 1 day realised volatility between 2014 - 2017 grouped by weekday (top left), hour of the day (top right), month (bottom left) and week of month (bottom right).

of the variables. This can be problematic for time series like the RV e.g. the training set between 2014 - 2017 is normalised, the same normalisation applied to a strong peak like Covid-19 would break the 0 and 1 balance. In other words, normalisation on the dataset can not account for unseen big peaks.

We went on to explore different transformations aimed at maximising performance. We decided to target logRV) as the variations in scale are squashed in logarithmic scale which naturally lead to more robust models against sudden jumps. We settled on transforming RV features into log(RV*10) and kept the moves unchanged. This transformation pushes the mean to zero which is crucial for the neural networks to perform effectively. We will return to the consequence of this transformation later in the results section.

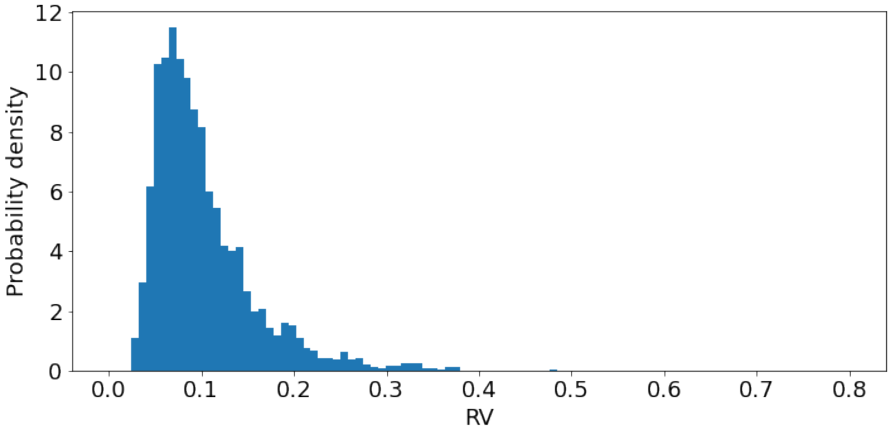

Figure 6.5: Normalised histogram of the rolling 1 day realised volatility from 2014-2017.

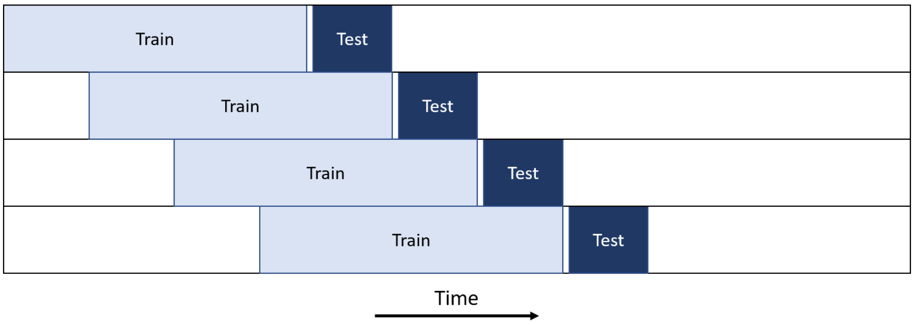

Figure 6.6: Training method of the benchmark model.

6.4 Methods

6.4.1 Benchmark

The benchmark method is a Ridge regression model trained with a walk forward window (rolling). The features used to train the benchmark is slightly different to the features mentioned in the previous section. Note that there is a tiny gap between the train and test set, this is to avoid overlapping labels as mentioned in the previous section. This method has strong transparency on how the weights are interacting at each re-train which allows for further analysis on how the model would react under different scenarios.

6.4.2 Artificial neural network (ANN)

We propose two types of neural network (NN) in predicting RV, artificial neural network (ANN)[1] and Bayesian neural network (BNN). The architecture we used for the ANN included 2 hidden dense layers each with 32 nodes, ReLU activation and L2 activity regulariser with $\lambda = 10^{-5}$. We did not use sigmoid because of the vanishing gradient problem where predictions on big peaks performed poorly. Each hidden layer is followed by a dropout layer with their rates set to 0.3. There are three general types of regularisers and they are kernel, bias and activity regularisers. The activity regulariser penalise on the combined weight between the kernel and the bias.

Validation is included through validation split where we use the last 30% of the training set as validation data. We have allowed random shuffling on the training data to improve on generalisation. The batch size is set to 100 and the number of epochs is set to 20. We have also employed model checkpoints and early stopping, a checkpoint is updated when the validation loss from the current epoch is the historical minimum and the model weights from this epoch get saved out. Early stopping monitors a given metric and terminates the training process when a condition is met. The metric we monitor is the validation loss and termination occurs when the validation loss remain unchanged or increased for 5 consecutive epochs. The trained models have their checkpoint weights loaded in before inference. The mean prediction from 200 models is used as the stable result. All of the models were trained with mean squared error as the loss function and Adam optimiser [45]. All DL models presented were implemented in `Tensorflow` [143].

6.4.3 Bayesian neural network (BNN)

The premise of BNNs is analogues to ANNs as mentioned in Section 3.3.4. The main differences between the two comes in the type of layer and the definition of

[1] ANN here is the same type as the fully connected neural network described in Section 3.3.3

loss function. For the hidden layers, BNN typically uses dense flipout layers instead of standard dense layers. Flipout layers perform a Monte Carlo approximation of the distribution integrated over the kernel and bias [53]. In other words, the kernel and bias are not points but distributions. The weight updates are now on the mean and standard deviation of the surrogate distributions which we have assumed are Gaussian.

The output layer could be used like in an ANN where the network directly predicts the target with 1 output node. The other way is to make an assumption about the distribution of the target variable and the network would then predict the mean and standard deviation of this distribution. We know the target distribution is a right skewed Gaussian with a fat tail from Fig. 6.5. Unfortunately, skewed distributions are still in development when this is written. Therefore, we used BNN as if it is an ANN with an uncertainty band.

The loss function is critically different with the inclusion of the Kullback-Leibler divergence (KL) on top of just MSE. In order to perform inference on a BNN, we need a method to optimise the distributions of the weights and bias. KL-divergence measures the similarity between two probability distributions and in this case between the prior distribution and the Monte Carlo approximated distribution. In addition, this extra term naturally acts as a regulariser and therefore BNNs are much less likely to overfit. A complete derivation of this process can be found in [55].

In practice, these convenient features of BNN means that we do not need to train an ensemble of models. However, this is not the full story because we need to sample the weight and bias distributions enough times to get a representative result. We found sampling 200 time to be sufficient and we take the mean as the stable prediction. This process is also known as variational inference.

BNNs were implemented with `Tensorflow` and `Tensorflow probability` [167]. The architecture used for the BNN is essentially the same as the ANN with the dropout layers removed. The number of epochs has moved up to 100 as BNN generally requires more epoch before convergence. The model checkpoint and early

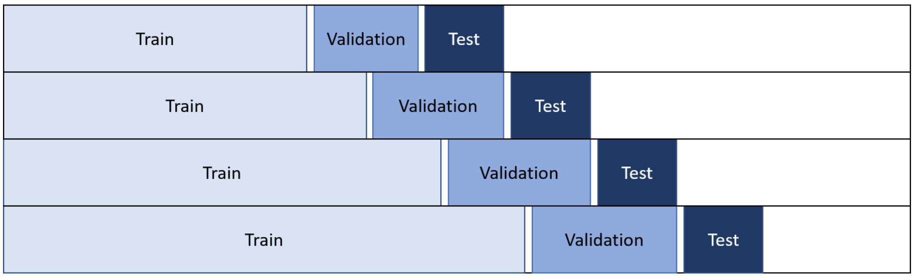

Figure 6.7: The anchored walk-forward training scheme.

stopping remain in place with the patience setting also increased to 10 epochs. All other hyperparameters are the same as the ANN.

6.4.4 Anchored walk-forward training

This is much like the walk forward training scheme used by the benchmark but the starting point of the training set is fixed while the whole training set increases in size like an expanding window as shown in Fig. 6.7. The first training set starts from 2014-2017 and the test set is 5 days long. The test set get absorbed to be the end point of the next training set. The validation data does not have a fixed size because it is always the last 30% throughout all timesteps. We have also tried to apply the exact same walk forward scheme from the benchmark to the neural networks but the result was bad. This is because the training sample size is simply too small for the networks to learn anything meaningful under the same specification of the benchmark.

6.4.5 Cross-validation

In the neural networks mentioned, they both have a validation set obtained from the final 30% of the training data. The cross validation here is similar to K-fold cross validation in terms of how the data is split but we have also included an anchored walk-forward type training scheme. We have 2 ways of using this, the first being an epoch number finder and the second is basically K-fold cross validation. Consider 5 sets of training data separated by similar number of validation data points, each

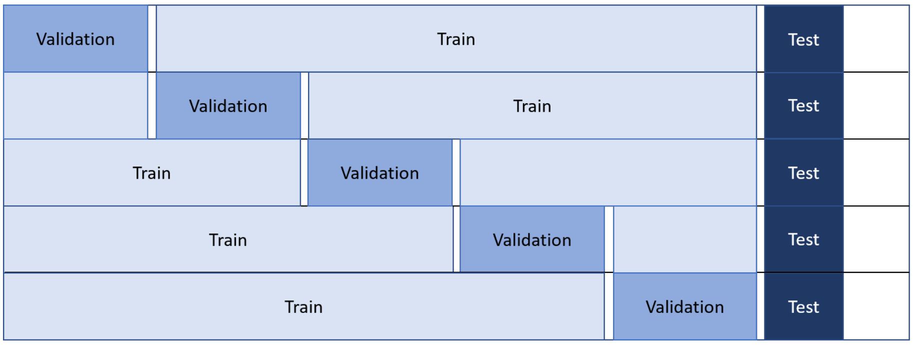

Figure 6.8: Cross validation by simultaneously training different sets one epoch at a time.

set has 50 models to account for instability. We train 250 models 1 epoch at a time. The mean validation loss (val loss) across all models is calculated, when this mean val loss is above the historical minimum val loss, the patience ticker goes up by 1. This ticker is reset to 0 if the current mean val loss is a new minimum. This training process stops when the patience reaches 5 and we would have an epoch number signalling the beginning of over-fitting. This is essentially a custom early stopping mechanism where the mean val loss across all validation sets is monitored. In the first scenario, we run the epoch finder every 2 months and use that number to train a fresh set of models without any validation data up to that epoch number every 5 days. The point is we now have a relatively safe epoch number where we could include the latest data in the training set without spending any of them on validation. The second usage is much simpler where we would use the 250 models as a big ensemble and take the mean prediction at every timestep. The train test split setup is displayed below in Fig. 6.8.

6.4.6 EMASE

This is an evaluation method intended to act as a safety net for models to combine their predictions in a conservative way. Consider an ensemble S of 2 models where one of them is an ANN and the other one is the benchmark model. We only used 2 models but there is no limit on the total number. The ANN generally performs well but ANNs are known to be bad at extrapolation i.e. it might give bad results if the data input does not belong to any part of the trained data space. On the other hand, the Ridge regression trained like the benchmark will always be reasonable due to its interpretability.

For each point of the prediction, the contribution from each of the models are weighted. The weights are calculated through the Error Function Moving Average Squared Error (EMASE) [168], it can be written as:

$$\text{EMASE} = \frac{\text{Erfc}(\text{MASE}_s)}{\sum_{s \in S} \text{Erfc}(\text{MASE}_s)}, \forall s \in S \ , \tag{6.4.1}$$

where S is the ensemble, s is each one of the models, MASE is the moving average square error over a period P and Erfc is the Gaussian complementary error function given in Eq. 6.4.2. The period P of the MASE controls the reactiveness such that smaller window leads to faster reactions. Note that faster reaction does not guarantee better predictions. We chose P to be the past 5 days such that we are sure about the weights without being too conservative. The MASE are then MinMax transformed to be between 0 and 1 with the minimum and maximum within every P. This retains the sense of scale between different predictions.

$$\text{Erfc}(x) = \frac{4}{\sqrt{\pi}} \int_x^\infty \exp(-t^2) dt \ . \tag{6.4.2}$$

The sum of EMASE adds up to 1, each weight decays exponentially with the loss due to the error function. The final prediction is then the linear combination between the weights and their respective predictions.

6.4.7 Model weight regularisation

This is an incremental learning method similar to the walk-forward training scheme from the benchmark. Rather than retraining the models at every timestep, we have a set of base models trained with data from 2014-2017. They have the same architecture as the ANN presented above. The new models have a similar architecture but the dropout layers are removed. At every timestep, we fine tune the new models with the base model weights loaded in with the hidden layer weights frozen such that the output layer is the only trainable object. The new models are trained on the most recent 30 days of data and test on the next day. The loss function is also custom made to include a penalty term on the mean squared difference between the base model weights and the fine tuned model weights. The custom loss is written as:

$$\text{Custom loss} = MSE + \frac{\lambda}{N}(w_{\text{base}} - w_{\text{FT}]})^2 \quad , \tag{6.4.3}$$

where MSE is the mean squared error, λ is the penalty coefficient much like the one for Ridge or LASSO, N is the length of the weights vector, w_{base} is the vector of base model weights and w_{FT} is the vector of the fine tuned model weights. This penalty is small when the fine tuned prediction is similar to the base prediction i.e. the fine tuned model weights are only allowed to change when the fine tuned prediction needs to be drastically different from the base prediction. This could happen when the input data is unseen within the base trained data space hence this method has a built-in safety net. This method introduced an additional parameter λ, we found $\lambda = 2 * 10^5$ gives the best result through calibrating on 2018-2019 data. One can also include an additional penalty term on the bias between the two sets of models to further constrain any changes. However, this additional degree of freedom makes it difficult to determine the best λ for each of the terms.

6.5 Result and discussion

The results shown in Table 6.1 were calculated on 2018-2019 data, the metrics included mean squared error (MSE), R-squared (R^2) and mean absolute error (MAE). In this table, standard refers to the statically trained base model, WF stands for walk forward, CVs are the corresponding usages mentioned in the previous section and fine tuned is the incrementally trained weights regulated model. Notice that there isn't a BNN fine tuned model, this decision was made because it is not reasonable to take weights differences from samples of the weight distributions. However, one could attempt to use the mean and standard deviation for such regularisation.

	Method	MSE	R^2	MAE
Benchmark	walk-forward	0.00240	0.566	0.03093
ANN	Standard	0.00205	0.630	0.02894
	Anchored WF	0.00203	0.633	**0.02877**
	Anchored CVWF epoch	0.00208	0.624	0.02938
	Anchored CVWF mean	0.00206	0.629	0.02882
	EMASE	0.00212	0.617	0.02926
	Fine tuned	0.00206	0.627	0.02905
BNN	Standard	**0.00195**	**0.648**	0.02916
	Anchored WF	0.00207	0.626	0.02898
	Anchored CVWF epoch	0.00208	0.625	0.02976
	Anchored CVWF mean	0.00196	0.645	0.02885
	EMASE	0.00202	0.635	0.02913

Table 6.1: Result calculated based on 2018-2019 data.

Overall, all methods performed better than the benchmark. The best fitted method is the standard BNN with $R^2 = 0.648$. The best ANN is the anchored walk forward model which also has the best MAE. Given the fact that big differences between prediction and target are enlarged in MSE comparing to MAE, the similarity in MAE across the models suggests that the predictions are all quite similar apart from the amplitude of the bigger peaks. When comparing the same method between ANNs and BNNs, the latter generally have higher R^2 especially the standard, anchored CVWF mean and EMASE models. This comes from better peak amplitude predictions as R^2 is proportional to MSE.

The first 4 months' predictions for each of the families are shown in Fig. 6.9 and 6.10. It is apparent that DL methods are quicker at reacting to sudden market movements as all models showed an earlier rise than the benchmark for the first big peak. As mentioned above, most predictions within the same family are quite similar apart from the anchored CVWF epoch finder models. They have likely still overfitted in certain regions as the number of epochs is decided through a relatively small batch size and the mean validation loss across all data up till the point of testing. It is also clear that BNNs generally predicted the peak amplitude with greater accuracy as mentioned before. Note that the benchmark predicted a false signal around mid

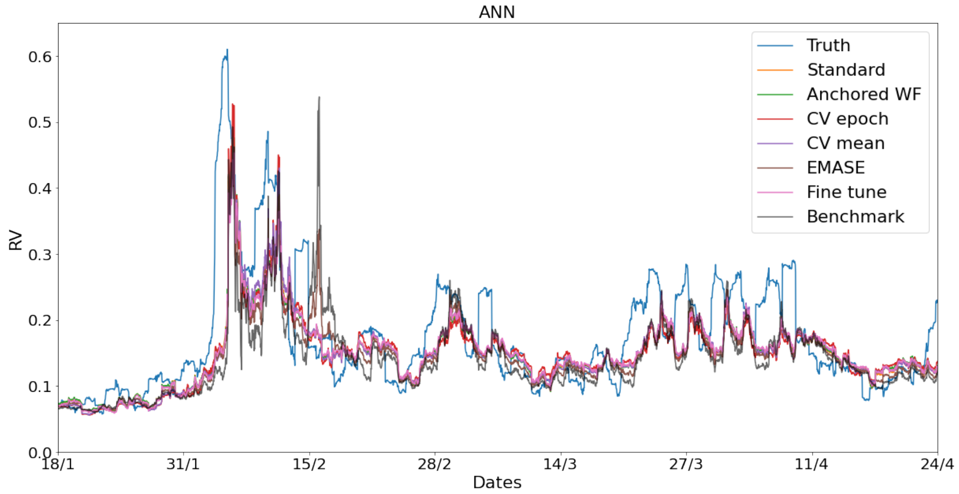

Figure 6.9: ANN predictions on the first 4 months of 2018 for each of the methods shown in Table 6.1.

February as a result of the choice of features. The EMASE predictions were strongly affected by it and have a damped version of this false signal as expected.

It is important to show that a strategy is profitable before applying it in production. We utilised the backtester developed by the Statarb team to test the trading signals generated from our models. The backtester simulate trades based on historical market movement. The 2018-2019 backtest results are shown in Table 6.2, Sharpe here refers to the annualised Sharpe ratio, ac shown in the bracket is the Sharpe ratio after cost and max drawdown is the biggest drop in PnL from any single trade throughout the testing period. Note that all models apart from the stared one has been tuned to have around 40% market time in short positions and 15% market time in longs. We will therefore not include the stared result in our performance comparison.

From Table 6.2, we can see that the simpler ANNs generally traded better than the other models. The standard ANN made the highest total PnL even though it did not have the best fitting result. The standard BNN which fitted best performed well on shorts but made a few bad longs. This is because of the higher peak amplitude where it would signal for longs even when the IV is on its way down. This is also known as a loss in Vega PnL. On the other hand, the best fitted ANN actually

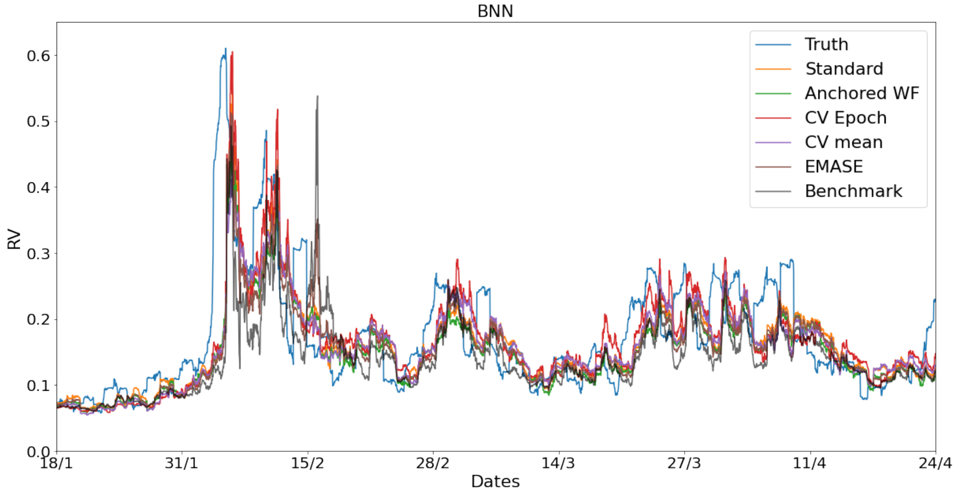

Figure 6.10: BNN predictions on the first 4 months of 2018 for each of the methods shown in Table 6.1.

traded well with the best Sharpe before cost on both shorts and longs. In terms of drawdowns, the BNN EMASE model came out on top but this only means that it has lost the least in any single trade. It is also the most profitable BNN. Regardless of the weak performance from the standard BNN, this shows the value in having a safety net. The safety net effect is not as prominent for the ANN EMASE because the standard ANN is much more similar to the benchmark than the standard BNN. Therefore the EMASE calculated often stays around 0.5 for the ANN.

In general, 2018 had some strong long opportunities due to explosion in VIX and other general market adjustments where else 2019 is mainly a short year because the RV had a downward trend throughout the year apart from occasional events. The ideal strategy would then be one that picked up these key features. The plots in Fig. 6.11 and 6.12 shows the evolution of the PnL for the respective methods. Note that the effect of VIX explosion and Trump tweet have been toned down to 0.2 of the original strength. All models roughly followed the ideal trait. It is interesting to see that the standard and the anchored CV epoch mean BNNs perform so similarly to the benchmark overall in total PnL while their fits are 15% better in terms of R^2.

	Method	Side	Count	Sharpe (ac)	PnL	Max drawdown
Benchmark	walk-forward	short	93	3.434 (2.604)	1.167	-0.217
		long	120	2.909 (1.633)	0.647	-0.109
ANN	Standard	short	99	4.898 (3.972)	1.710	-0.128
		long	142	4.088 (2.587)	**0.872**	-0.121
	Anchored WF	short	86	**5.311** (**4.405**)	**1.749**	-0.123
		long	139	**4.482** (2.514)	0.706	-0.125
	Anchored CVWF epoch	short	85	4.149 (3.266)	1.374	-0.127
		long	125	3.304 (1.752)	0.583	-0.108
	Anchored CVWF mean	short	85	4.083 (3.209)	1.367	-0.123
		long	183	3.622 (1.446)	0.639	-0.138
	EMASE	short	94	4.132 (3.230)	1.344	-0.162
		long	133	3.818 (2.149)	0.702	-0.095
	Fine tuned	short	90	4.971 (4.006)	1.539	-0.111
		long	125	4.035 (2.616)	0.823	-0.133
BNN	Standard	short	98	4.052 (3.047)	1.338	-0.129
		long	60	2.172 (1.177)	0.359	-0.171
	Anchored WF *	short	64	5.605 (4.697)	1.515	-0.105
		long	119	4.371 (2.431)	0.544	-0.091
	Anchored CVWF epoch	short	83	4.886 (3.875)	1.447	-0.111
		long	58	3.379 (2.578)	0.603	-0.100
	Anchored CVWF mean	short	91	4.242 (3.311)	1.427	-0.143
		long	66	2.401 (1.179)	0.310	-0.180
	EMASE	short	96	4.867 (3.874)	1.496	**-0.119**
		long	70	3.946 (**2.949**)	0.703	**-0.095**

Table 6.2: Backtest result calculated based on 2018-2019 data.

6.5.1 2020

2020 is the out-of-sample test set as mentioned before. We have decided to only test the better models with this additional test set and they are the standard, anchored walk forward, EMASE, fine tuned ANNs and the EMASE BNN. The fitting results for these models are shown below: The benchmark predicted a massive peak in March at about $RV = 3$ when the truth is about 1.6 which explains the poor MSE and R^2. The rest of the predictions from the benchmark are reasonable. On the other hand, the NNs predicted fine apart from the region after the Covid-19 peak around April as shown below in Fig. 6.13. It is clear that the standard and anchored walk forward ANNs did not pick up the regime shift. They were constantly predicting 0.05 points below the benchmark let alone the truth. There are several reasons

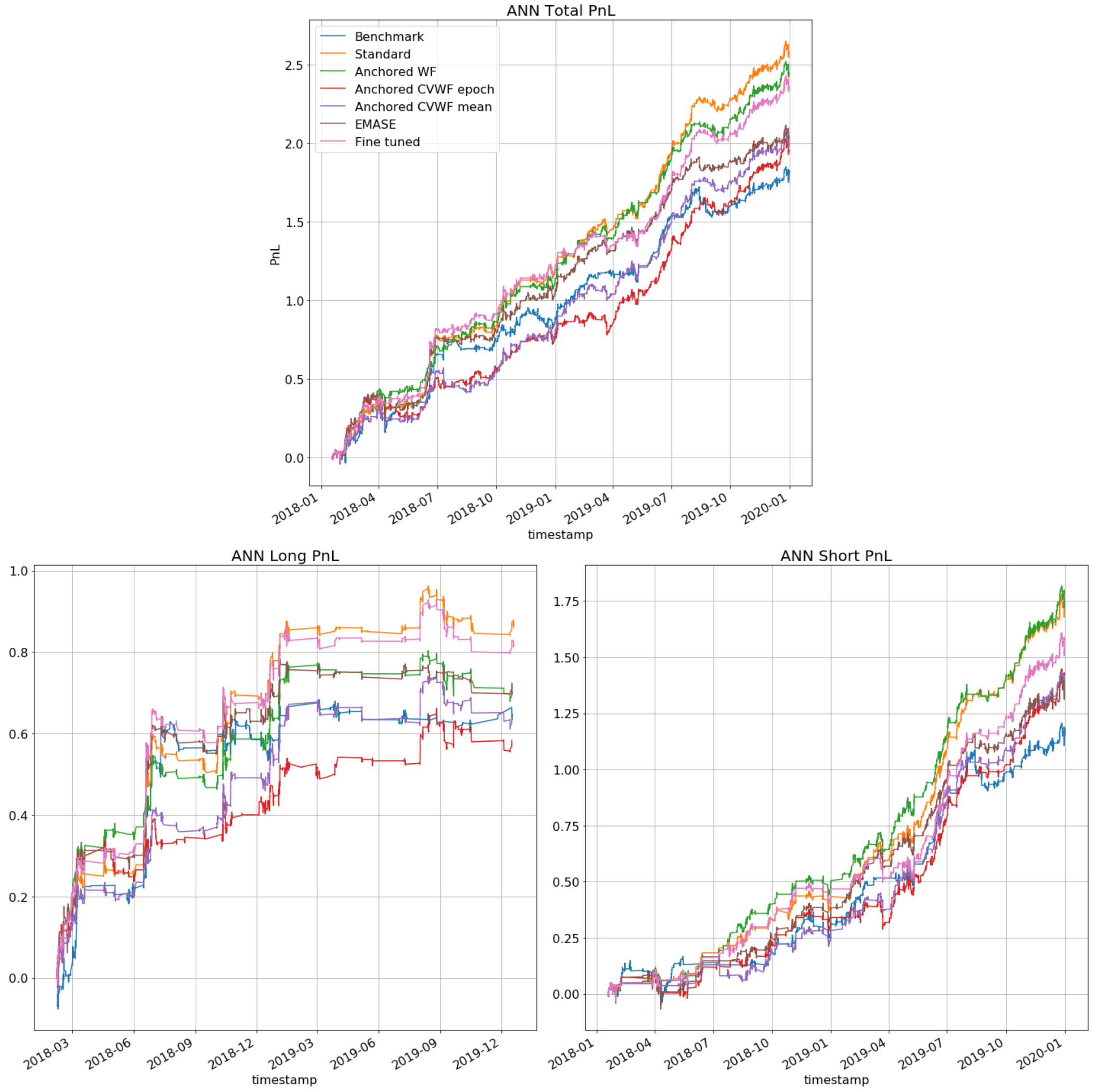

Figure 6.11: Backtest result of the ANNs, total PnL (top), PnL from longs (bottom left) and PnL from shorts (bottom right).

behind this, the first being the moves of the underlying during that time. The moves after the big peak in March have mostly been positive as the market sharply recovered from the enormous drop, the models have likely learnt that positive moves are related to decrease in volatilities and vice versa. Therefore, the predictions drops quickly to a similar level before Covid-19 where it is comfortable. Secondly, this is exactly the kind of bad extrapolation neural networks could make because there weren't any similar events between 2014-2017 in terms of market impact. Notice that the models with safety nets overcame this problematic region as designed, this

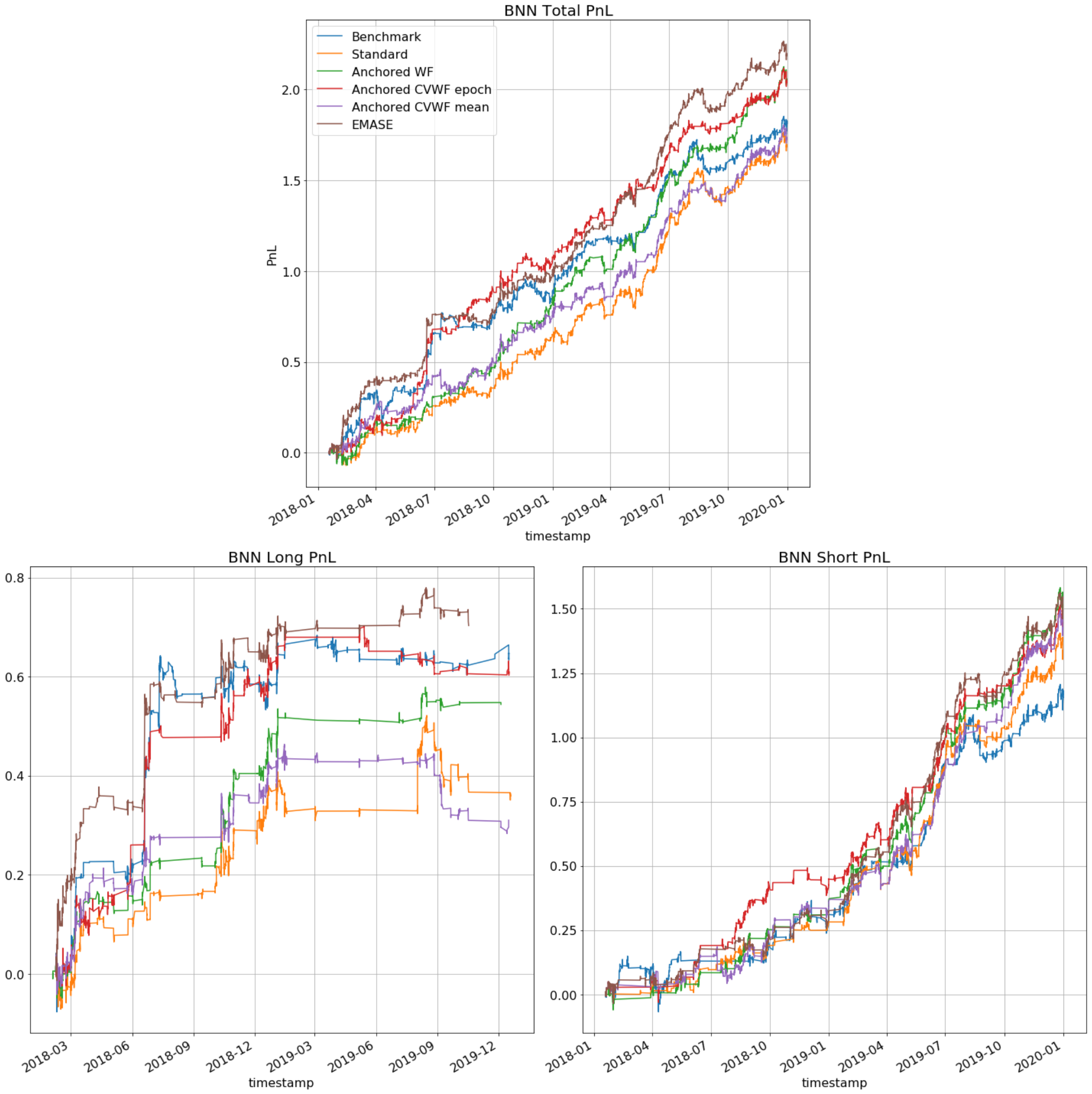

Figure 6.12: Backtest result of the BNNs, total PnL (top), PnL from longs (bottom left) and PnL from shorts (bottom right).

shows the importance of such mechanisms as the future is unpredictable. Lastly, this could be a curse from predicting in log scale where the predictions and labels are not significantly different in log space but this difference is actually quite big once transformed back for low RV. However, this particular problem in April should be avoidable given more training data as we know retrospectively the RV behaved similarly to other extreme events such as the 2008 financial crisis.

There were two major events in our 2020 data, the assassination of Qasem Soleimani and Covid-19. The RV has been on a downward trend at times other than these

	Method	MSE	R^2	MAE
Benchmark	walk-forward	0.04134	0.241	0.08357
ANN	Standard	0.01840	0.662	0.08158
	Anchored WF	0.02061	0.622	0.07970
	EMASE	0.01795	0.670	0.07289
	Fine tuned	**0.01604**	**0.705**	0.07282
BNN	EMASE	0.01699	0.688	**0.07130**

Table 6.3: Fitting result on 2020 data.

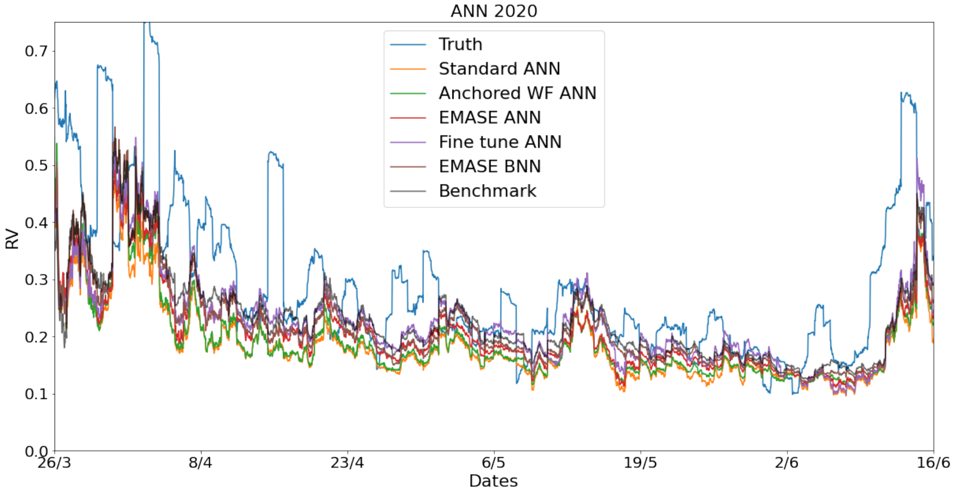

Figure 6.13: Predictions in April after the Covid-19 peak.

events making 2020 a fairly simple year in terms of trades. The backtest result on the first 9 months of 2020 is shown in Table 6.4 and Fig. 6.14. Note that the effect on the Covid-19 peak has also been toned down like previous major events. The trade barriers for these models remained the same as calibrated for 2018-2019, the market time in shorts are around 70% and 8% in longs.

The DL models all avoided shorting into the assassination while some actually longed the event and gained strong profit, this is most likely luck from datetime patterns. The overall performance between the benchmark and the DL models are not that different otherwise. In addition, none of the DL models got punished for predicting the wrong long term RV because IV was above the RV for pretty much the whole time past the peak. Notice that the short PnL dropped a few times while no longs

	Method	Side	Count	Sharpe (ac)	PnL	Max drawdown
Benchmark	walk-forward	short	46	1.330 (0.784)	0.339	-0.254
		long	20	8.909 (7.344)	0.072	-0.017
ANN	Standard	short	44	3.038 (2.437)	0.595	-0.126
		long	44	4.057 (1.856)	0.111	-0.033
	Anchored WF	short	44	**3.358** (**2.749**)	**0.695**	-0.136
		long	50	1.170 (-1.142)	0.030	-0.059
	EMASE	short	45	2.892 (2.293)	0.574	-0.156
		long	37	7.167 (5.750)	**0.221**	-0.019
	Fine tuned	short	46	3.013 (2.399)	0.579	-0.168
		long	41	4.186 (2.666)	0.130	-0.028
BNN	EMASE	short	46	2.726 (2.091)	0.525	**-0.136**
		long	17	**9.290** (**7.868**)	0.105	**-0.013**

Table 6.4: Backtest result from January to the end of September of 2020.

were made during the second half of the year. They were Covid-19 news and it is impossible for our kind of models to predict such jumps but it would be interesting to see if a headline scrapper type algorithm could help minimise the loss.

6.5.2 Extra data

Towards the end of this project, we obtained more S&P data dated from 2000. This data contains several major events such as the 2008 financial crisis and the 911 attack in 2001. The 2008 peak clearly shares similar regime shift type behaviour to the Covid-19 peak as shown in Fig. 6.15. We applied the standard ANN model training from 2000-2016 and test on 2017-2020, we can expect better performance from the model in 2020 because of the other events. Training on this bigger dataset has also improved performance in other test years as the network got to learn from more examples of datetime effect. We have excluded hour of the day from the datetime features as it was inducing false patterns, the rest of the feature set is the same as above. The prediction for April 2020 is displayed in Fig. 6.16, the new ANN performed exactly as desired solving the extrapolation problem. In addition, we compared the backtest results dated from 2017 onwards with the benchmark and

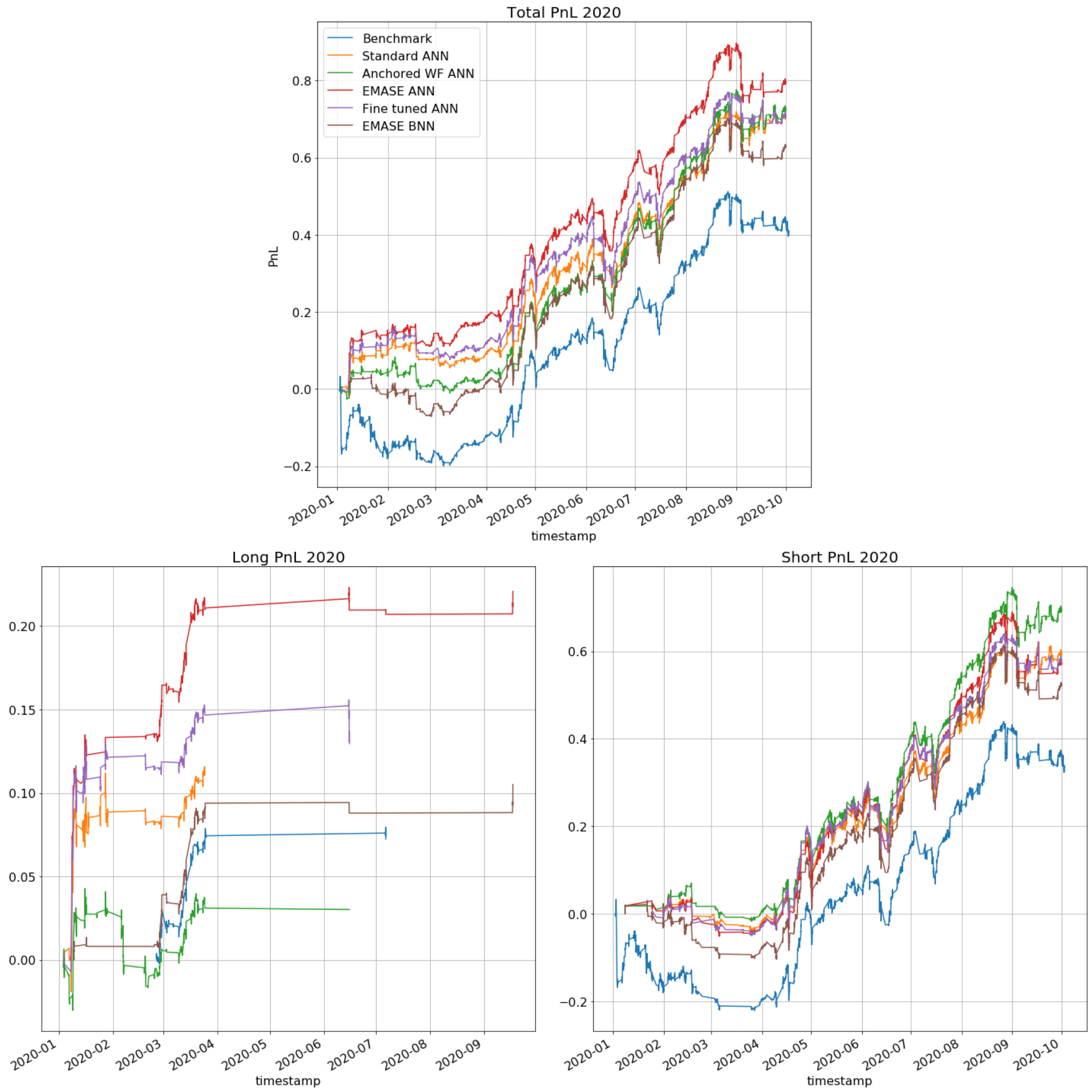

Figure 6.14: Backtest result of the selected DL models in 2020.

the standard ANN as shown in Table 6.5 and Fig. 6.17. There is a staggering 2 units of PnL difference between this ANN and the benchmark. This difference comes from longs in 2017 and late 2019. It is interesting how none of the previous models made similar longs in late 2019. Further investigation is required to gain better understanding on this part.

6.6 Summary

We have explored different DL methods to predict RV, they have mostly performed

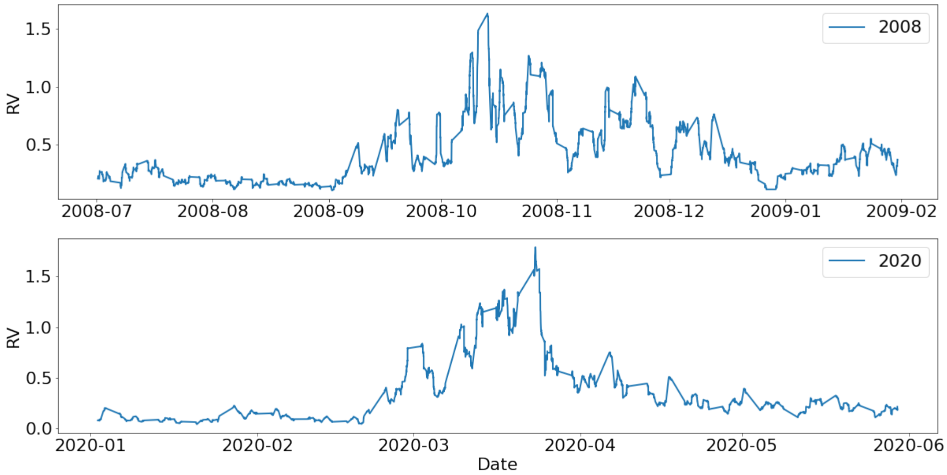

Figure 6.15: Comparison between the 2008 financial crisis peak and the Covid-19 peak in 2020.

	Method	Side	Count	Sharpe (ac)	PnL	Max drawdown
Benchmark	walk-forward	short	187	2.601 (1.797)	2.203	-0.583
		long	166	3.262 (1.724)	0.898	-122
ANN	Standard	short	266	4.679 (3.602)	3.636	-0.388
		long	172	4.705 (3.027)	1.523	-0.114

Table 6.5: Backtest result from 2017-September 2020.

better than the benchmark. The standard, anchored walk forward, fine tuned ANNs and the EMASE BNN have comparable results as the top candidates.

The success of these models demonstrated some key features, performance in R^2 does not necessarily translate to trading performance as shown by the standard BNN but this is not definite for every model as seen in the anchored walk forward ANN. This lower R^2 comes from lower predicted peak amplitude which seemingly helped avoid problematic longs for the better traded models. We have tested on more complicated models like LSTM or just with added complexity to the architecture without success. The simple architecture with a low number of neurons restricts the freedom of the models reducing the possibility of overfit and false signals. The models without safety nets suffered from the April regime shift problem in 2020 as they produce poor extrapolations. However, preliminary result from training with

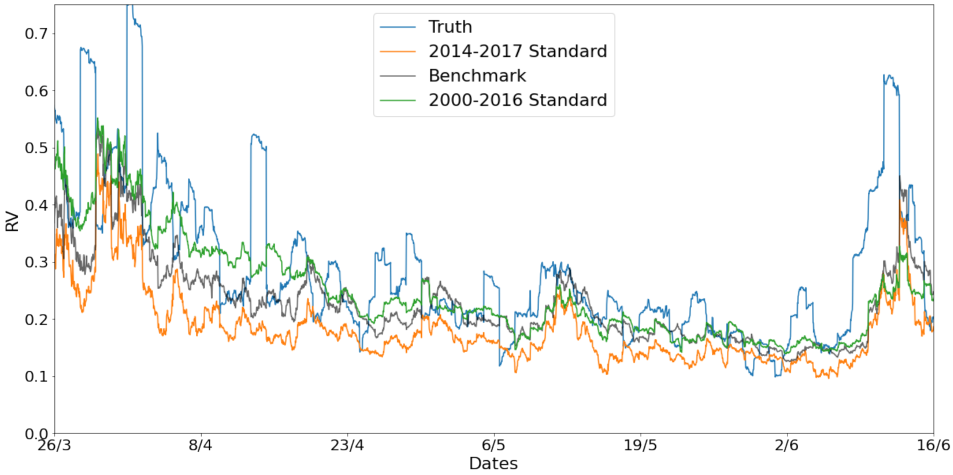

Figure 6.16: Comparison between standard ANNs trained with 2014-2017 data and 2000-2016 data in April 2020.

additional data showed that the April problem can be resolved by having events similar to Covid-19 in terms of market impact. The future of this project can be separated into 3 main directions, the first in architecture. We have not fully explored the capabilities of this new dataset, more complicated methods such as LSTM may work better than standard ANNs given the larger training dataset. The second one is on feature searches, the current feature set is rudimentary as we only have features from the return series and datetime effects. Other features from different underlying or insights from traders should be explored. The third direction would be to use alternative data like news headlines or something more extreme like satellite sensory data as an increasing amount of movement in the market can be traced back to certain instantaneous events.

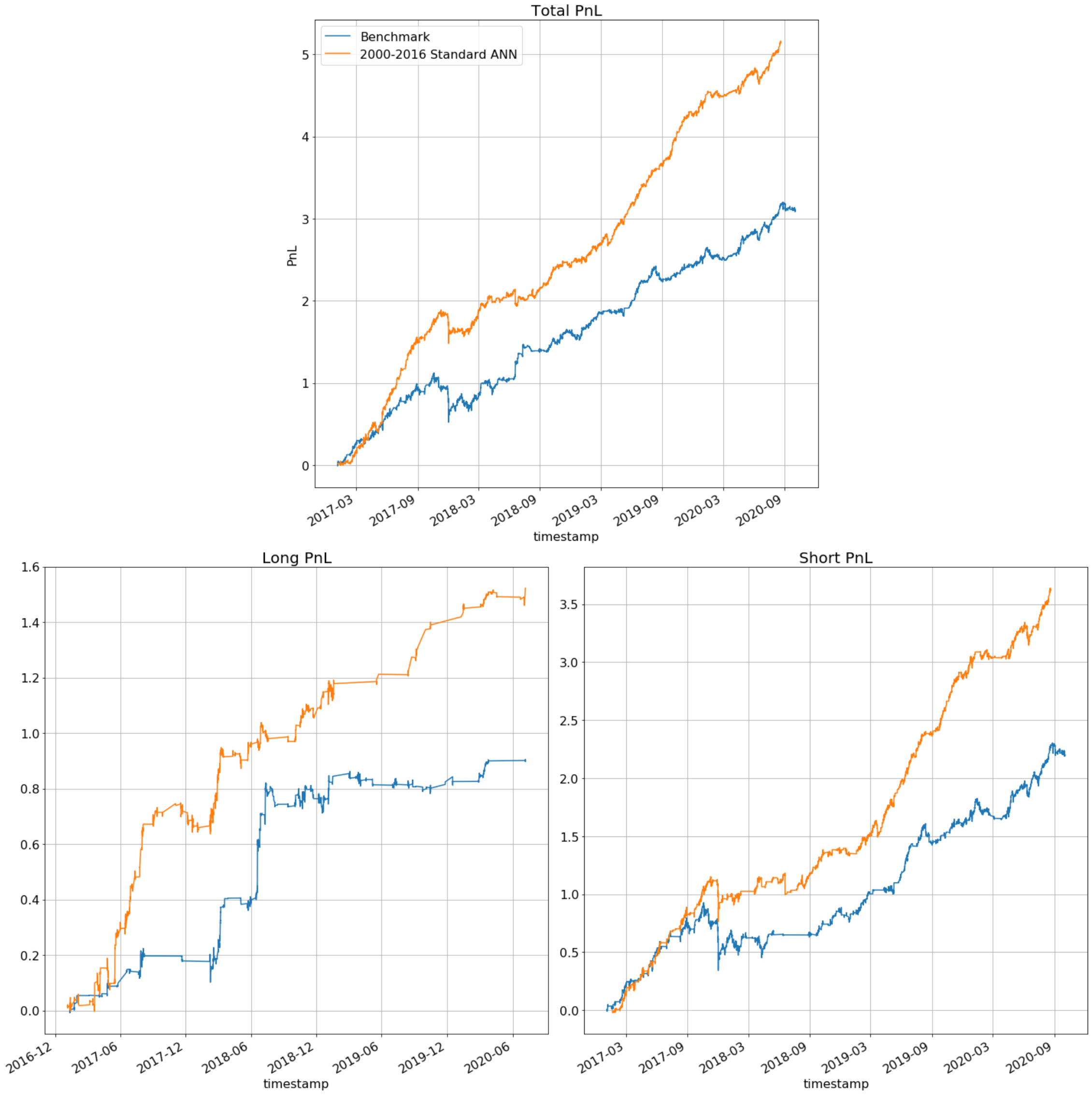

Figure 6.17: PnLs between the benchmark and 2000-2016 trained standard ANN, total PnL (top), long PnL (bottom left) and short PnL (bottom right).

Chapter 7

Conclusion

This thesis explored the usage of machine learning algorithms on two classification problems, inclusive $|V_{ub}|$ determinations and strange quark jet flavour tagging. Both problems focus on flavour physics as $|V_{ub}|$ is the least known element within the CKM matrix and strangeness tagging is the final piece of the puzzle in quark flavour tagging. A strong emphasis on feature selection is imposed for both problems, as advancements in data driven solutions allows for low level detector information in addition to the carefully crafted physics observables. The aim of the studies is then to provide new methods and features in extending development for these areas and in turn be better equipped when testing the flavour structure of the Standard Model (SM).

We began with an overview of the thesis in Chapter 1. We followed with an introduction to the SM and in particular flavour physics in Chapter 2. Chapter 3 focuses on descriptions of different machine learning methodologies and their deployment within particle physics.

In Chapter 4, we studied systemically the application of ML techniques to inclusive $|V_{ub}|$ determinations. We showed that a deep neural network trained on low level single-particle detector level features resulted in a small performance increase compared to the existing Boosted Decision Tree method based on high-level observables used in the Belle analysis [80]. The difference in performance being so small further validates the effectiveness of the carefully crafted high-level features in discriminating the $b \to u$ signal from the overwhelming $b \to c$ background.

In addition to method comparisons, we dived into understanding the inclusivity within the fiducial region selected by cuts on the classifier output of two types of neural networks. The networks differ in the input features respectively where one included both kinematic and hardon-level features and the other only used hadron-level features similar to the BDT approach mentioned. These signal acceptance rates provide insights on behaviours of the models when selecting signal events. They also prompt for additional consideration when extrapolating $|V_{ub}|$ as systematic theory uncertainties associated to the Monte Carlo modelling emerge for non-local shape function OPE regions.

This chapter finished with an investigation on the influence of $s\bar{s}$-popping probability for the signal acceptance rates as supervised ML approaches require precise MC data, especially when measurements become systematics dominated for Belle II. The full analysis was carried out using the standard $|V_{ub}|$ analyses MC tool EvtGen but was also repeated with a different event generator Sherpa. The NNs were trained and tested on data produced by different MCs as a stress test on MC uncertainties. Further opportunities arise from more complicated network architectures but also advancements in theory calculations such that MC generators can keep up with the precision required in future experiments.

In Chapter 5, we explored the potential of building a strangeness tagger among light quark jet backgrounds with NN architectures. We showed that the experimental jet definition has little effect as the quark-matched samples contain mostly leading jets ordered by p_T. We systematically studied the features, in particular J_s and its compositions [124, 141]. We observed the importance of particle identification through performance of the feature groups and that the network must have effectively created something similar to $p^{\parallel}$ and J_s from low level detector tracks information through a SHAP analysis.

Furthermore, we investigated the prospect of having such a network setup for LHCb.

We included the gluon contributions as part of the background events. The result suggests that the signal and background ML efficiencies are similar to the simpler s-jets vs d-jets scenario.

Finally as a side study, we investigated the effectiveness of obtaining PID information through time-of-flight measurements planned for general purpose detectors such as ATLAS and CMS. We showed that the proposed 30 ps time resolution is not precise enough for this individual setup and the required resolution is lower than 1 ps.

There are several directions forward for this study, one way is to employ more sophisticated network architectures. However, this also requires additional research on more effective flavour discriminants. Another limit is with PID where precise detection of charged kaons would only improve tagging performances.

Chapter 6 is a diversion from particle physics where a research project was presented in collaboration with Optiver to investigate the application of deep learning in predicting realised volatility of financial indexes. We examined two different neural network architectures along with numerous training schemes in terms of how the data is arranged to maximise performance. The benchmark is a simple ridge regression and all proposed DL methods achieved stronger results in both the fits and the backtesting result obtained from simulated trades. In addition, this study showed the importance of the quality of the training data, as a regime shift occurred at the beginning of 2020 (COVID-19), which was not picked up by models trained with data between 2014-2017. The reason behind the poor fit is because such behaviour has not been observed within that period and NNs are unreliable when extrapolation is required. However, when data dated from 2000 are included as part of the training data, similar market behaviour from the 2008 financial crisis helped the models overcome the underprediction by the model trained on 2014-2017 data.

As follow-ups to this study, the obvious next step is to further investigate the robustness of the presented methods. In terms of potential areas for improvement, the most important one is to search for more effective and/or diverse features. The current set of features are mostly derived from the returns of the underlying. Correlations between the features are naturally high but perhaps causality should also be considered as a proxy for different events. Diversity can also come from a different data source, an interesting choice would be news headlines, though it is difficult to incorporate such data as market reactions to news are extremely subjective.

Appendix A

$|V_{ub}|$ study appendices

A.1 Detector simulation

Theoretically, the signal and background processes are well separated by the through kinematic boundaries at $M_X = m_D$, $P_+ = m_D^2/m_B$ and $p_\ell^* = (m_B^2 - m_D^2)/(2m_B)$. However, detector effects lead to large contributions from the $B \to X_c\ell\nu$ background to the $B \to X_u\ell\nu$ signal region, and it is necessary to include them in order to mimic the challenges of the experimental environment.

In the following, we describe our in-house detector simulation meant to capture the main features of a more complete one. We list the assumed parameters for detector resolution in Section A.1.1 and for detector efficiencies and mistagging probabilities in Section A.1.2. Most of these values are based on the description of the BaBar detector in Ref. [169], from the BaBar analysis of the inclusive determination of $|V_{ub}|$ paper [81] and the corresponding PhD thesis on the same subject [170]. We compare the resulting distributions after our detector simulation to those shown in the recent reanalysis of Belle events in Ref. [82]. We highlight that the beam energies in Belle (3.5 GeV and 8.0 GeV) are slightly different from the values we used in our MC event generation (4.0 GeV and 7.0 GeV), see Section 4.3. We therefore expect deviations of the lab-frame momenta on the level of $\lesssim 10\,\%$.

A.1.1 Detector resolution

We assume perfect reconstruction of the direction of each detected particle and we only smear the energy (momemtum) for photons (charged particles). The energy resolution of photons is parametrized by [170]

$$\frac{\sigma_{E_\gamma}}{E_\gamma} = \frac{2.32\,\%}{E_\gamma^{1/4}} \oplus 1.85\,\%, \quad E_\gamma \text{ in GeV.} \tag{A.1.1}$$

For the resolution of charged particles, we use the p_T resolution of the Drift Chamber (DCH) which is the main tracking device for charged particles with $p_T \geq 120\,\mathrm{MeV}$ [170].

$$\frac{\sigma_{p_T}}{p_T} = 0.45\,\% \oplus 0.13\,\%\,p_T, \quad p_T \text{ in GeV.} \tag{A.1.2}$$

We apply this formula on all charged particles, also those with $p_T < 120\,\mathrm{MeV}$.

A.1.2 Efficiencies and mistagging

For charged particles/tracks, the overall reconstruction efficiency is 98 % for momenta $p \geq 200\,\mathrm{MeV}$ (DCH) [170]. We assume that mistagging is only relevant for

$$\begin{aligned} \text{true } \pi^\pm &\rightarrow \text{ fake } K^\pm & \qquad \text{true } K^\pm &\rightarrow \text{ fake } \pi^\pm \\ &\rightarrow \text{ fake } e & &\rightarrow \text{ fake } e \\ &\rightarrow \text{ fake } \mu & & \end{aligned}$$

Photons

Photons are detected with an efficiency of 96 % for energies above 20 MeV.

$$\mathrm{eff}_\gamma(E_\gamma) = 0.96\,(E_\gamma \geq 0.02), \quad E_\gamma \text{ in GeV} \tag{A.1.3}$$

Electrons

Electrons need to have a minimum momentum of $p_{\text{lab}} = 500\,\text{MeV}$ in the lab frame. Their efficiency is 93 % above this threshold [81].

$$\text{eff}_e(p) = 0.93\,(p \geq 0.5), \quad p \text{ in GeV} \tag{A.1.4}$$

Muons

Muons need to have a minimum momentum of $p_{\text{lab}} = 500\,\text{MeV}$ in the lab frame. Their efficiency is 90 % above this threshold.

$$\text{eff}_e(p) = 0.9\,(p \geq 0.5), \quad p \text{ in GeV} \tag{A.1.5}$$

Since muons and electrons/hadrons are detected in different detector parts, we assume the muon fake rate for electrons and hadrons to be negligible.

Kaons

Charged kaons need to have minimum momenta of $p_{\text{lab}} \geq 300\,\text{MeV}$ to be identified. The efficiency is taken from Fig. 3.5 of Ref. [170]. It drops linearly for momenta satisfying $p < 7\,\text{GeV}$, at values above this we approximate the efficiency using a quadratic function:

$$\text{eff}_{K^\pm}(p) = \begin{cases} 0, & p < 0.3 \\ -0.8\,p + 1.23, & 0.3 \leq p < 0.7 \qquad p \text{ in GeV} \\ 0.86 - 0.35(p-1.5)^2, & 0.7 \leq p < 1.8 \\ -0.0225\,p + 0.87, & p > 1.8 \end{cases} \tag{A.1.6}$$

We determine possible K_s^0 candidates based on the invariant mass of opposite-sign pion pairs. Pairs in the mass range $m_{\pi^+\pi^-} \in [0.490,\, 0.505]\,\text{GeV}$ are assumed to result from K_s^0 decays with a 40 % probability, see Fig. 3.6 of Ref. [170]. We model the misidentification of kaons as electron as

$$\mathrm{mis}_{e|K}(p) = \begin{cases} 0, & p < 0.05 \\ 0.004 - 0.001\,p, & 0.05 \leq p < 4.0 \qquad p \text{ in GeV} \\ 0, & p > 4.0 \end{cases} \tag{A.1.7}$$

Pions

For the reconstruction efficiency of slow, i.e. low momentum, pions we use the values given in Ref. [171]. The efficiency for pions grows exponentially from 20 % at $p_T = 50\,\mathrm{MeV}$ to 80 % at $p_T = 70\,\mathrm{MeV}$, see also Fig. 9 of Ref. [171]. For pion momenta $p \geq 0.4\,\mathrm{GeV}$, we assume the reconstruction efficiency to drop linearly, compare Fig. 89 of Ref. [169].

$$\mathrm{eff}_{\pi}(p) = \begin{cases} 0, & p < 0.05 \\ 1 - 13\exp(-86.29\,p + 560.4\,p^2 - 1601\,p^3 + 1625\,p^4), & 0.05 \leq p < 0.4 \qquad p \text{ in GeV} \\ 1 - 0.015\,p, & p > 0.4 \end{cases} \tag{A.1.8}$$

The efficiency for pions to be misidentified as kaons is taken from Fig. 3.5 of Ref. [170]. We approximate the momentum dependence as linear for low momenta and constant for larger momenta

$$\mathrm{mis}_{K|\pi}(p) = \begin{cases} 0, & p < 0.05 \\ 0.01\,p, & 0.05 \leq p < 2.0 \qquad p \text{ in GeV} \\ 0.02, & p > 2.0 \end{cases} \tag{A.1.9}$$

We assume the efficiency for pions to be misidentified as muons to be 0.5 % below 1 GeV and 1 % above this value (Fig. 3.4 in Ref. [170]). We do not model any angular dependence.

$$\mathrm{mis}_{\mu|\pi}(p) = \begin{cases} 0, & p < 0.5 \\ 0.005\,p, & 0.5 \leq p < 1.0 \qquad p \text{ in GeV} \\ 0.1, & p > 1.0 \end{cases} \tag{A.1.10}$$

We model the misidentification of pions as electron as

$$\mathrm{mis}_{e|\pi}(p) = 0.001p\,(p > 0.5), \quad p \text{ in GeV} \tag{A.1.11}$$

A.1.3 Validation

To validate our detector simulation, we reproduce Fig. 14 of Ref. [82] in our Fig. A.1. We find good agreement for the number of charged kaons and the bulk of the $M^2_{\mathrm{miss},\,D^*}(\pi_{\mathrm{slow}})$ distributions. Larger deviations between our detector simulation and the Belle values, for instance at low $M^2_{\mathrm{miss},\,D^*}(\pi_{\mathrm{slow}})$ or with a large number of kaons, appear in statistically much less relevant regions and less than 2 % (1 %) of all signal (background) events lie at $M^2_{\mathrm{miss},\,D^*}(\pi_{\mathrm{slow}}) < -20\,\mathrm{GeV}^2$. Less than 3 % of the background event contain more than one charged kaon. Since we do not include the effect of particles from the tagging side of the event being assigned to the signal side, we poorly underestimate the negative regime of the missing mass squared.

A.2 Machine Learning analysis set-up

A.2.1 Training and test sets

To train our classifiers, we create balanced data sets with 10M $B \to X_u \ell \nu$ signal events and 10M $B \to X_c \ell \nu$ background events. The data preparation process includes the application of the in-house detector simulation and a standard scaling of the data based on the training set. Categorical features are one-hot encoded and are not scaled. The training set is shuffled and 20 % of it is used for cross validation. For testing, we create two test sets with a physical signal-to-background ratio (1/45). Each test set contains 40K signal and 1.8M background events after detector simulation, which roughly corresponds to the number of semi-leptonic B-decays in a sample of 22.6M $B\bar{B}$ events.

A.2.2 Bayesian neural network

Bayesian neural network (BNN)	
Input layer	number of features nodes
1^{st} hidden DenseFlipout layer	256 nodes, Sigmoid activation batch normalisation
2^{nd} hidden DenseFlipout layer	256 nodes, Sigmoid activation batch normalisation
3^{rd} hidden DenseFlipout layer	256 nodes, Sigmoid activation
Output layer	1 node, Sigmoid activation
Kernel posterior function	mean field normal distribution
Bias posterior function	mean field normal distribution
Kernel divergence function	KL divergence function
Loss function	binary cross-entropy
Optimizer	Adam
Batch size	512
learning rate	0.1 for first 10 epochs then decreasing with $e^{-0.1}$ each epoch

Table A.1: Neural network architecture.

Our BNN is implemented with `Tensorflow` [143], `TensorFlow-Probability` [167] and `Keras` [145] with a total of 5 layers. The number of nodes of the input layer is the number of input features. There are 3 hidden `DenseFlipout` layers [53], each of them containing 256 nodes using the Kullback-Leibler (KL) divergence function as the kernel divergence function. We use a sigmoid activation function for all hidden layers. The first two hidden layers are followed by a batch normalisation layer which scales the weights and biases to have mean = 0 and standard deviation = 1. This helps avoid the vanishing gradient problem with sigmoid functions. The output layer only has 1 node with a sigmoid activation function, the posterior function for the kernel and bias are both assumed to be mean field normal distributions. The kernel divergence function for the output layer is also the KL divergence function.

We use binary cross-entropy as our loss function and apply the Adam [45] optimizer. The KL divergence is automatically added to the loss during training. Early stopping and model checkpoints are in place to monitor the validation loss of each epoch. The model weights from the best performing epoch are saved out and loaded back in before inference. We summarise the NN architecture in Table A.1.

A.2.3 Boosted decision tree

Boosted decision tree (BDT)	
Classifier	XGBoost
Max depth	10
Learning rate	0.4
Number of estimators	300
Gamma	1
Subsample	0.9
Colsample_bytree	0.7
Loss function	logloss

Table A.2: Boosted decision tree architecture.

The BDT is implemented with `XGBoost` [40]. We allow for a maximum depth of 10 as higher depth did not improve performance. The learning rate is fixed at 0.4. The number of estimators is set to 300 with early stopping in place. The gamma factor is fixed at 1. The subsample ratio of the training instance is 0.9 and subsample ratio of columns when constructing each tree is set to be 0.7 to reduce the risk of overfitting. The BDT set-up is summarized in Table. A.2.

In training the algorithms, the hyperparameters displayed in Tab. A.1 and A.2 were predetermined with minimal optimization through `HyperOpt` [146].

A.3 Plots of the high-level input features

A.4 Training with Sherpa

In Section 4.5 we studied the performance of $\mathrm{NN}_{\mathrm{tight}}$ and $\mathrm{NN}_{\mathrm{loose}}$ when trained on EvtGen data and then tested on both Sherpa and EvtGen data. Here we give results when instead Sherpa data is used to train the BNNs.

We begin by showing in Fig. A.3 the signal acceptances of $\mathrm{NN}_{\mathrm{tight}}$ (upper row) and

$\mathrm{NN}_{\mathrm{loose}}$ (bottom row), finely binned in the variable M_X. As in Fig. 4.8, the plots also show the signal and total number of accepted events (TP+FP), normalized to the number detector-level signal events, in addition to the background acceptances for $\mathrm{NN}_{\mathrm{loose}}$ using the y-axis shown on the right of the lower panels. The plots in the left-hand side of the figure are trained on EvtGen data, while those on the right are trained on Sherpa data.

The figure shows that the signal acceptances for $\mathrm{NN}_{\mathrm{tight}}$ are fairly independent of the training and testing data up until about $M_X \sim 1.5$ GeV, even though finely-binned signal modelling from the two MCs is vastly different. For $M_X > 1.5$ GeV, on the other hand, the acceptances depend crucially on the which MC is used in the training. The reason is that the Sherpa signal drops quickly to zero beyond this point, and is already negligible at the D-meson resonance at $M_X = 1.9$ GeV. Consequently, as seen in the top-right plot, a Sherpa-trained $\mathrm{NN}_{\mathrm{tight}}$ tends to reject the higher-M_X region of the EvtGen signal, as it has not seen signal events in that region during the training.

This artificial separation of signal and background in Sherpa is an unphysical effect that can be remedied by a matching with OPE-based results, which give a model-independent description of fully inclusive rates in the higher-M_X region. We note further that the signal acceptance of $\mathrm{NN}_{\mathrm{loose}}$ is fairly flat as a function M_X, whether trained on EvtGen or Sherpa data, and in particular even the Sherpa-trained version accepts EvtGen signal events across the entire region. In this case, however, the unphysical behaviour of the signal modelling would inevitably show up in a poor fit quality in the second stage of the analysis. For these reasons we have not considered Sherpa-trained BNNs in the body of the text.

Still, for completeness, we show in Fig. A.4 and A.5 the Sherpa-trained versions of Fig. 4.8 and 4.10. The most prominent feature is the expected reduction in the signal acceptance of EvtGen data by $\mathrm{NN}_{\mathrm{tight}}$ in the regions of high-M_X and low q^2 and p_ℓ^* in in Fig. A.4 compared to the EvtGen-trained version in Fig. 4.8, as well as a higher acceptance of the Sherpa signal overall, regardless of the BNN.

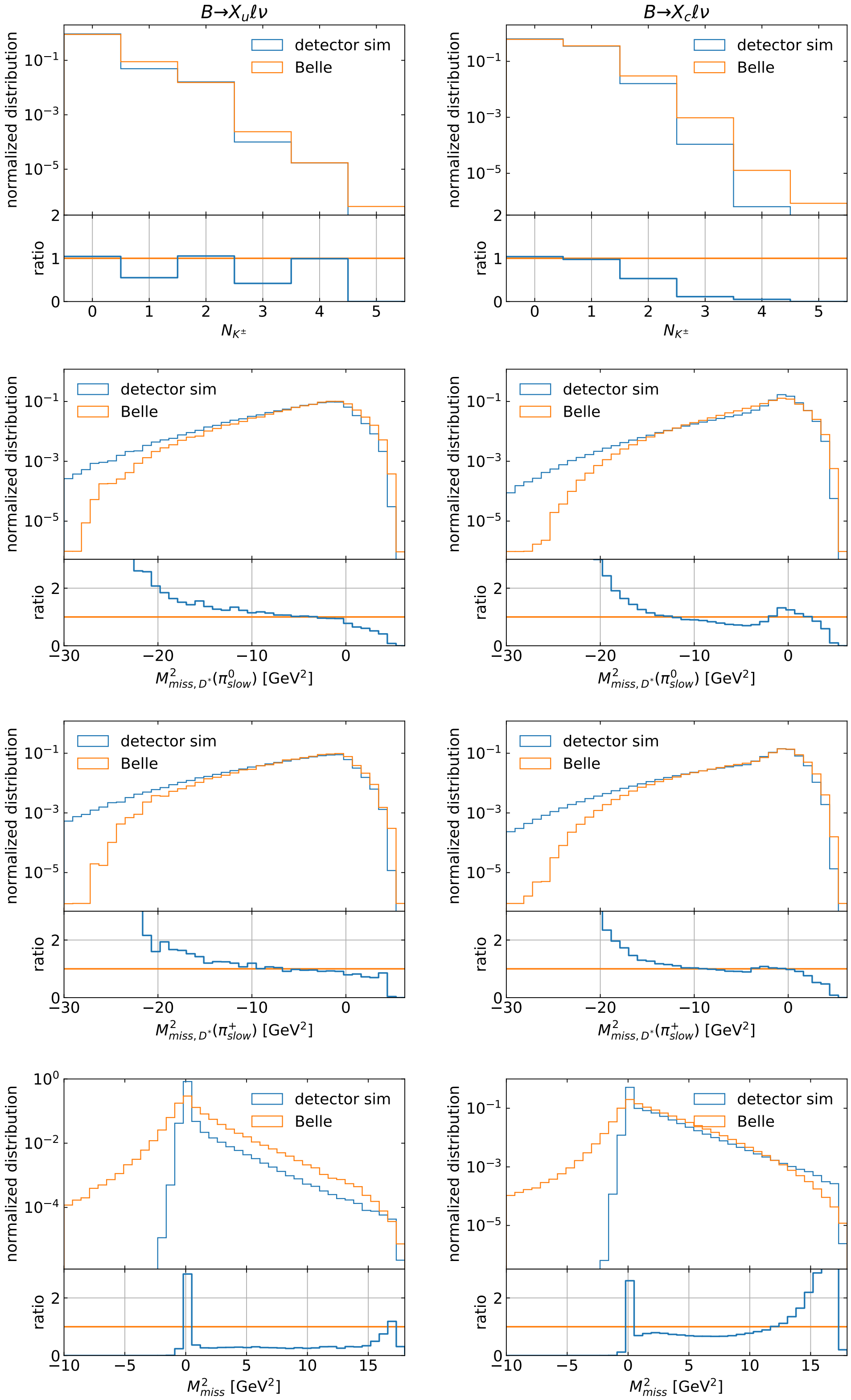

Figure A.1: Detector simulation validation plots for signal (left) and background (right) contributions. We compare the distributions of our MC events after detector simulation (detector sim) with the MC events produced by the Belle collaboration displayed in Fig. 14 of Ref. [82]. See paragraph below Eq. 4.4.2 for the feature definitions.

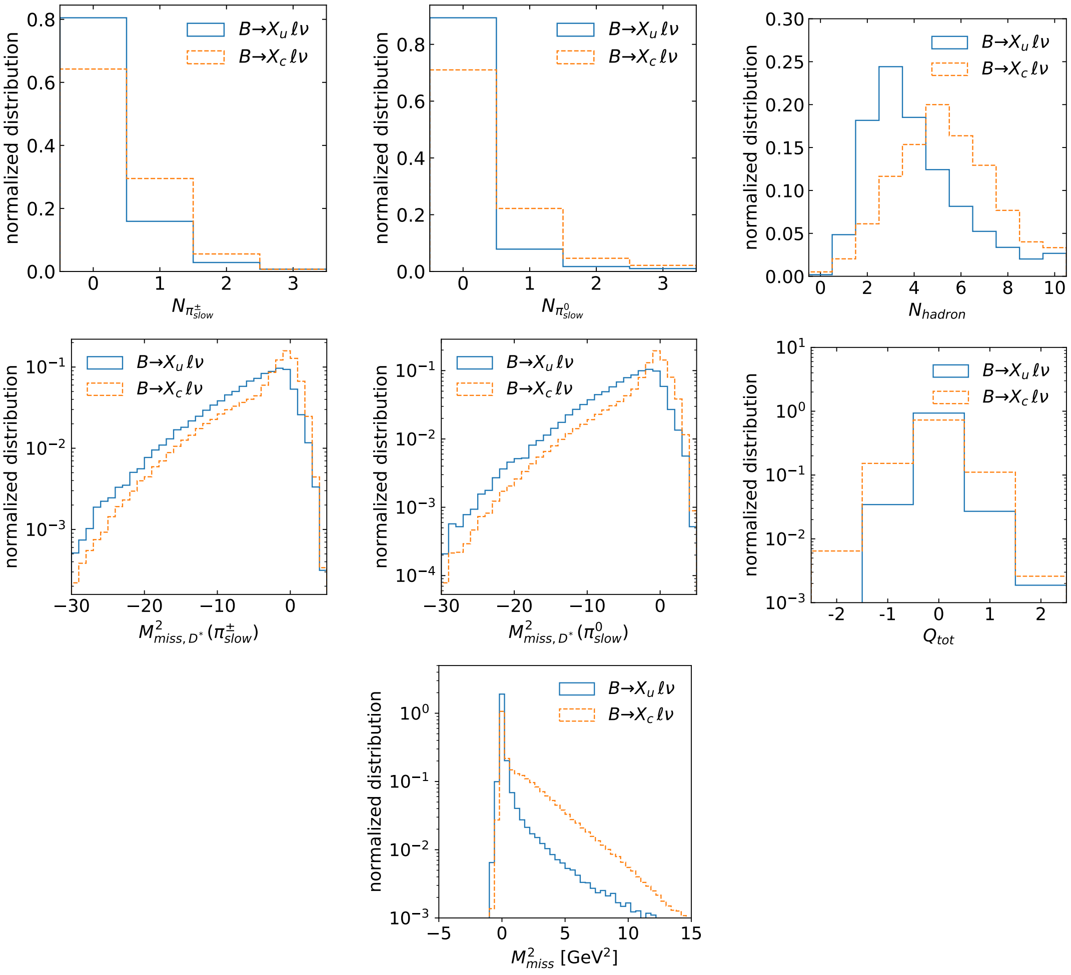

Figure A.2: Comparison of high-level features for $B \to X_u \ell\nu$ signal and $B \to X_c \ell\nu$ background events.

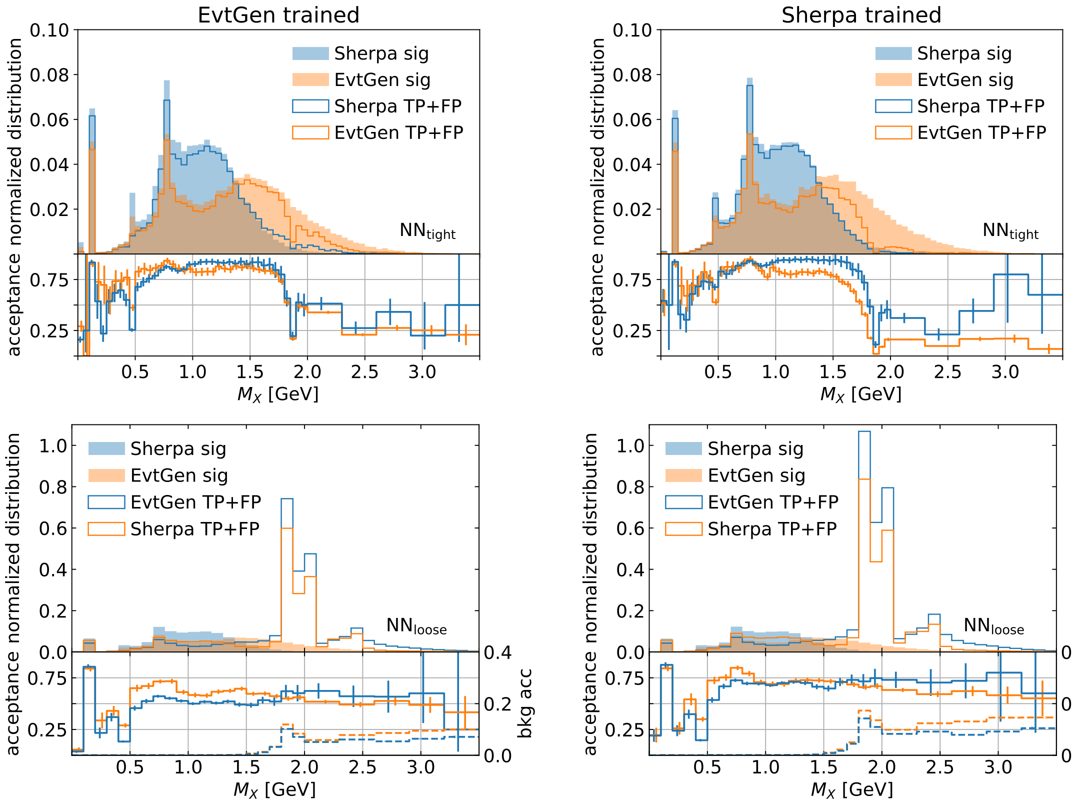

Figure A.3: M_X distributions and signal acceptance for NN_tight (top) and NN_loose (bottom) trained on EVTGEN (left) and SHERPA (right) data. For NN_loose the dashed lines in the lower panel show the background acceptance using the scale for the y-axis on the right.The distributions in the upper panels of each plot are normalized to the total number of signal events. A broader binning has been chosen to show the acceptance at $M_X > 2\,\text{GeV}$, where event statistics are low.

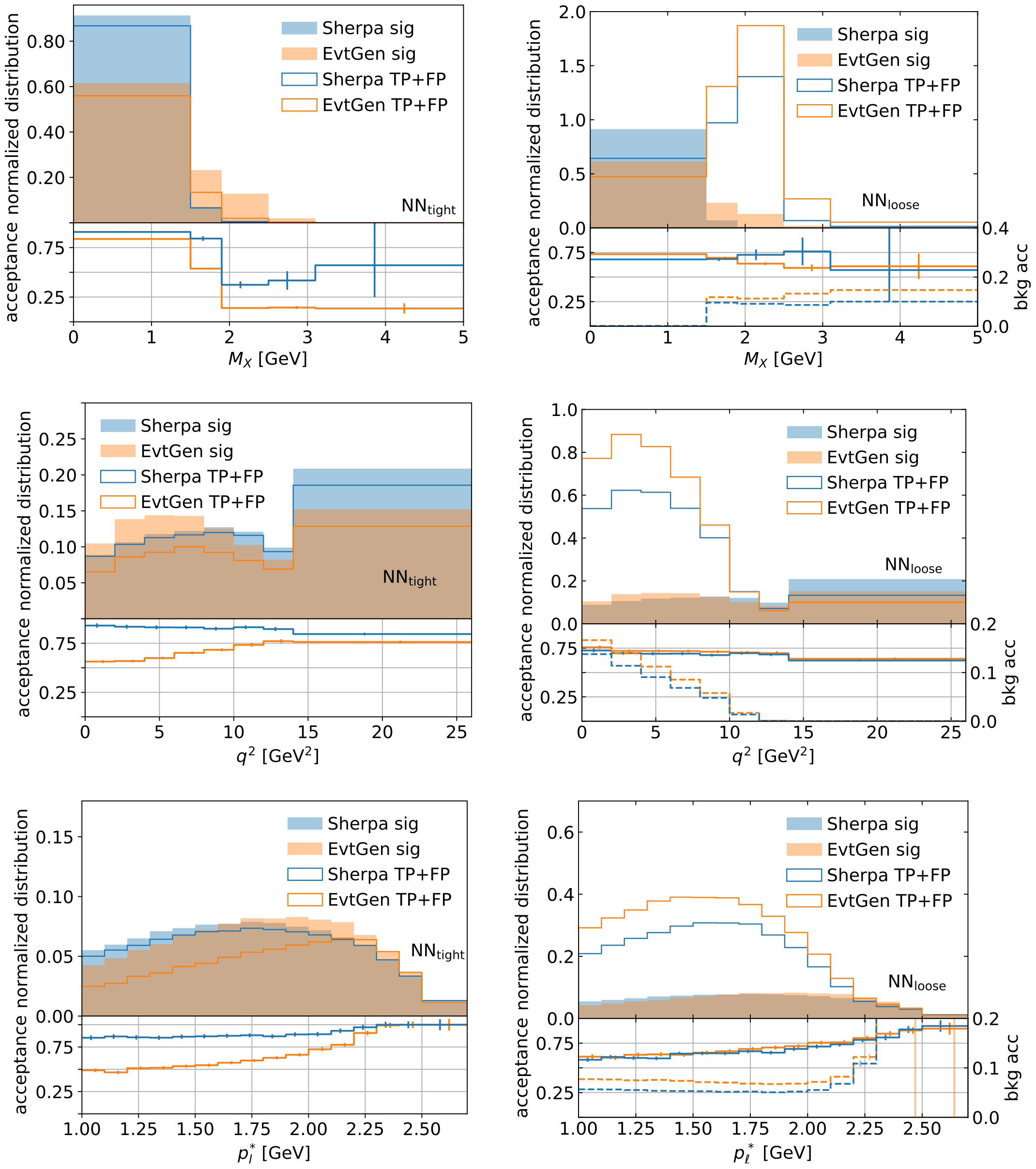

Figure A.4: As in Fig. 4.8, but using SHERPA instead of EVTGEN data for training the NNs.

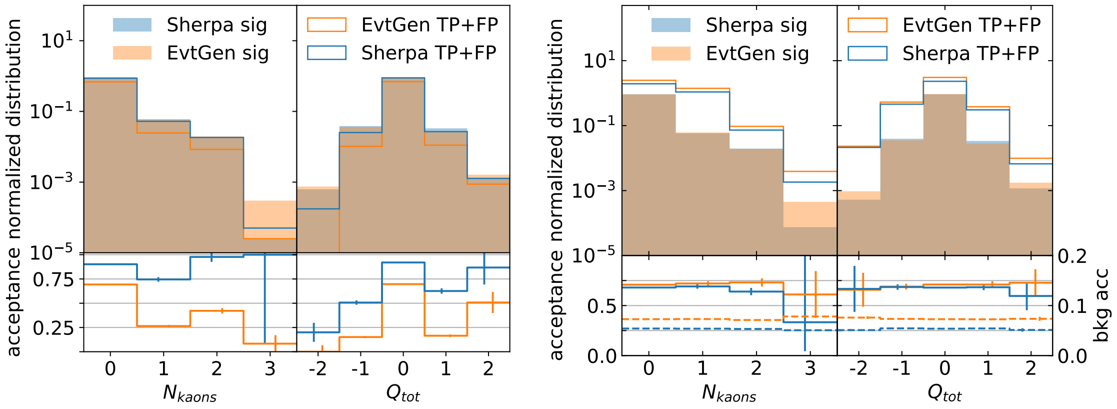

Figure A.5: As in Fig. 4.10, but using SHERPA instead of EVTGEN data for training the NNs.

Appendix B

Strangeness tagger appendices

B.1 Detector simulation specifications

In order to smear jet objects in the SFS [139] module we first choose constituent based smearing. The detection of partons is based on calorimetric energy and spatial resolution. The standard deviation in energy resolution has been taken as 10.32%, and the standard deviation in the spatial resolution has been determined by

$$\sigma_{\text{spatial}} = \frac{\alpha}{1 + e^{\beta(p_T^i - \gamma)}} \ ,$$

where p_T^i is the transverse momentum of the hadron and $\alpha,\ \beta$ and γ are taken as $0.04526,\ 0.013$ and 31.15, respectively which are based on Ref. [172]. Both energy and spatial resolution have been used in a Gaussian smearing function to set the momentum of hadrons. Furthermore, a reconstruction probability has been set on jets where the efficiency is defined as [173]:

$$\varepsilon_j = \begin{cases} 0\% & \text{for } p_T < 0.8 \text{ or } \eta < 2 \text{ or } \eta > 5 \ , \\ 95\% & \text{else} \, . \end{cases} \tag{B.1.1}$$

In addition to jets, electron (muon) reconstruction efficiency have been set as [174]

$$\varepsilon_{e\ (\mu)} = 0\% \text{for } p_T < 0.6\ (0.8) \text{ or } \eta < 2 \text{ or } \eta > 5 \ , \tag{B.1.2}$$

$$97\%\text{else}\,. \tag{B.1.3}$$

The electron and photon energy has been smeared according to the calorimeter specifications if they are within $2 < \eta < 5$,

$$\frac{\sigma_E}{E} = 1 \oplus 0.015E \oplus 0.1\sqrt{E}\ . \tag{B.1.4}$$

The track reconstruction efficiency has been held the same as jets where their momentum have been smeared with $\sigma_{p_T} = 0.005$ if their transverse momentum is greater than 0.5 GeV and its transverse impact parameter has been smeared with $\sigma_{d_0} = 0.0116 + 0.0234/p_T$. Since our simulation largely depends on particle identification, we also implement track misidentification where charged pions, kaons and protons are accepted with only 95% probability. The misidentification of pions, kaons and protons as muons has been held via

$$\begin{aligned} \varepsilon_{\pi|\mu} &= & 0.005 + 0.0663\ e^{-0.13\ p_T \cosh(\eta)}\ , \\ \varepsilon_{K|\mu} &= & 0.005 + 0.086\ e^{-0.11\ p_T \cosh(\eta)}\ , \\ \varepsilon_{p|\mu} &= & 0.2\%\ , \end{aligned}$$

respectively [173].

Alongside the smearing, the trajectory of each particle has been modified in accordance to 1.1 T magnetic field and 3.31 m tracker radius. Note that a similar parametrization has also been used in Delphes package [175].

B.2 Neural network architectures

This section contains descriptions for the two neural network architecture used in Chapter 5. The networks are implemented within TensorFlow version 2.1 [143, 144] and Keras [145]. All hyperparameters for the networks are optimised using `hyperopt` version 0.2.5 [146].

Large Neural Network (NN)	
Input layer	number of features nodes
1^{st} hidden Dense layer	16 nodes, Tanh activation, L2 regulariser = 1e-3
1^{st} Dropout layer	rate = 0.001
2^{nd} hidden Dense layer	128 nodes, ReLU activation, L2 regulariser = 1e-2
2^{nd} Dropout layer	rate = 0.06
3^{rd} hidden Dense layer	64 nodes, Sigmoid activation, L2 regulariser = 1e-2
3^{rd} Dropout layer	rate = 0.2
Output layer	1 node, Sigmoid activation
Loss function	binary cross-entropy
Optimizer	Adam
Batch size	256

Table B.1: NN architecture for models with numerous features.

Small Neural Network (NN)	
Input layer	number of features nodes
1^{st} hidden Dense layer	16 nodes, Tanh activation, L2 regulariser = 1e-4
1^{st} Dropout layer	rate = 0.235
2^{nd} hidden Dense layer	16 nodes, ReLU activation, L2 regulariser = 1e-4
2^{nd} Dropout layer	rate = 0.318
Output layer	1 node, Sigmoid activation
Loss function	binary cross-entropy
Optimizer	Adam
Batch size	256

Table B.2: NN architecture for high level only model.

B.3 Performance in general purpose detectors

In Section 5.5.2, we demonstrated the importance of PID capabilities. This section explores the application of the presented method to general purpose detectors. We use time-of-flight (TOF) as a proxy for PID at different time resolutions. The ATLAS and CMS collaboration have investigated the usage of TOF for identifying charged particles through the time-of-arrival from around 30 picosecond (ps) resolution [129, 130]. TOF by definition is the time it takes for hadron H to travel to the barrel of timing detectors from the collision point, it is given as :

$$\text{TOF} = \frac{R_b E_H}{p_T^H} = R_b \cosh \eta_H \sqrt{1 + \frac{m_H^2}{|\vec{p}_H|^2}} \tag{B.3.1}$$

where R_b is the distance from the point of collision to the barrel of the detector, c is the speed of light. Eq B.3.1 implies the mass m_H and therefore the identity of the charged final state particle can be extracted given we know the hadron 3-momentum (p_H). However, we must assume imperfect knowledge of TOF due to the timing resolution. We model detector resolution by smearing the truth level TOF with Gaussian noise of width t_{res}, where $t_{res} \approx 30$ ps represents the experimental situation. The mass squared (m^2_{TOF}) can be extracted as:

$$m^2_{TOF} = p^2_{T,H}\left(\frac{\mathrm{TOF}^2}{R_b^2} - \cosh^2\eta\right) \tag{B.3.2}$$

so that $m_{T,H} = m_H$ for $t_{res} = 0$, corresponding to perfect PID. For non-zero t_{res}, $m_{T,H}$ is distributed about m_H in a p_T-dependent way. This is especially important for high-p_T hadrons, where $m^2_H \ll p^2_{T,H}$. For such hadrons, the difference of the two terms inside the square root in Eq. B.3.2 must be much smaller than the terms themselves, so smearing the TOF makes a big effect on the reconstructed mass $m_{T,H}$. To see this, let us shift TOF → TOF + δt, where $\delta t \sim t_{res}$. Under this shift, $m_{T,H} \to m_H + \delta m_{T,H}$, where, setting $\cosh\eta = 1$ and working to first order in δt for simplicity,

$$\frac{\delta m_{T,H}}{m_H} = \frac{c\delta t}{R_b}\frac{p^2_{T,H}}{m^2_H}\sqrt{1 + \frac{m^2_H}{p^2_{T,H}}}\,. \tag{B.3.3}$$

The shift in the reconstructed mass is thus enhanced by p^2_T/m^2_H. Numerically,

$$\frac{\delta m_{T,H}}{m_H} \approx 0.8 \times \left(\frac{\delta t}{30\mathrm{ps}}\right) \times \left(\frac{p_{T,H}}{5\mathrm{GeV}}\right)^2 \times \left(\frac{m_K}{m_H}\right)^2\,. \tag{B.3.4}$$

Let us now ask what timing resolution δt is needed for $\delta m_{T,H}/m_{T,H} < 0.5$ at $p_T = 5$ GeV. For a kaon, this requires $\delta t \sim 20$ ps, but for a pion the number is $m^2_K/m^2_\pi \approx 12$ times smaller, $\delta t \approx 2$ ps.

The data for this additional study on general purpose detectors is also generated from SHERPA as ql^+l^- and $\bar{q}l^+l^-$ separately at $\sqrt{s} = 13$ TeV. Only $q = [s, d]$ were generated for this section. In addition, the MC specifications are roughly the same as described in Section 5.4 apart from the jet $|\eta| < 3$. Furthermore, detector effects

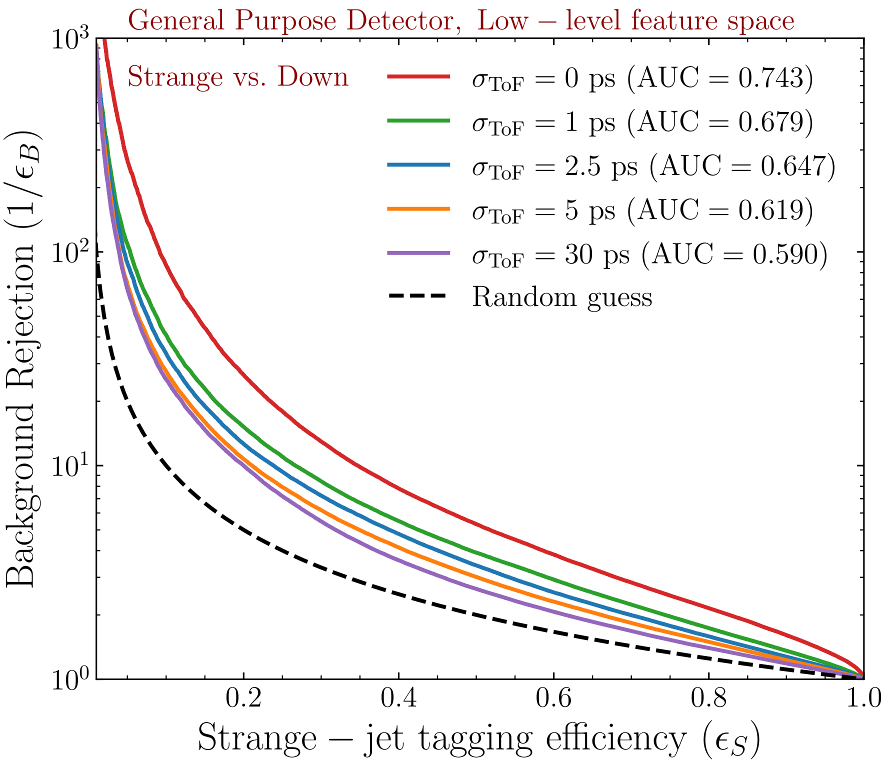

Figure B.1: ROC plot for s-jets vs d-jets in a general purpose detector with time-of-flight detector time resolution at [0, 1, 2.5, 5, 30] ps and mass resolution at 15% .

on particle misidentification and momentum smearing have been removed to solely focus on effects from the time-of-flight resolution.

We used the same NN architecture as described in Table B.1 for this study. The features are the same as the low-level features listed in Table 5.2. There is a slight difference in how the PIDs are obtained. They are derived from the time-of-flight mass shown in Eq. B.3.2 and hence have a dependence on the TOF. There is an additional parameter introduced to set an acceptance mass range for the PIDs, this mass resolution is set to be 15%.

In Fig. B.1, the ROC curves for time resolution $= [0, 1, 2.5, 5, 10, 30]$ ps are displayed. The result shows that the perfect 0 ps case perform similarly to the LHCb case. The proposed 30 ps under the setup described in this analysis is not feasible as the AUC performed poorly compare to the 0 ps case. The performances for other time resolutions are also disappointing as slight improvement is only observed when the time resolution reaches 1ps. There is a small caveat with this study which is the mass resolution. It is currently set at 15% but it is essentially a tunable hyperparameter through additional optimisation.

Bibliography

[1] K. Albertsson, P. Altoe, D. Anderson, J. Anderson, M. Andrews, J. P. A. Espinosa et al., *Machine learning in high energy physics community white paper*, 2019.

[2] M. E. Peskin and D. V. Schroeder, *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995.

[3] Particle Data Group collaboration, P. A. Zyla et al., *Review of Particle Physics*, *PTEP* **2020** (2020) 083C01.

[4] S. Marzani, G. Soyez and M. Spannowsky, *Looking inside jets*, *Lecture Notes in Physics* (2019) .

[5] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. Abdel Khalek, A. Abdelalim et al., *Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc*, *Physics Letters B* **716** (Sep, 2012) 1–29.

[6] S. Chatrchyan, V. Khachatryan, A. Sirunyan, A. Tumasyan, W. Adam, E. Aguilo et al., *Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc*, *Physics Letters B* **716** (Sep, 2012) 30–61.

[7] H. Fritzsch, M. Gell-Mann and H. Leutwyler, *Advantages of the color octet gluon picture*, *Physics Letters B* **47** (1973) 365–368.

[8] D. J. Gross and F. Wilczek, *Ultraviolet behavior of non-abelian gauge theories*, *Phys. Rev. Lett.* **30** (Jun, 1973) 1343–1346.

[9] H. D. Politzer, *Reliable perturbative results for strong interactions?*, *Phys. Rev. Lett.* **30** (Jun, 1973) 1346–1349.

[10] S. L. Glashow, *Partial-symmetries of weak interactions*, *Nuclear Phys.* **22** (2, 1961) .

[11] S. Weinberg, *A model of leptons*, *Phys. Rev. Lett.* **19** (Nov, 1967) 1264–1266.

[12] A. Salam and J. C. Ward, *Weak and electromagnetic interactions*, *Il Nuovo Cimento (1955-1965)* **11** (1959) 568–577.

[13] F. Englert and R. Brout, *Broken symmetry and the mass of gauge vector mesons*, *Phys. Rev. Lett.* **13** (Aug, 1964) 321–323.

[14] P. W. Higgs, *Broken symmetries and the masses of gauge bosons*, *Phys. Rev. Lett.* **13** (Oct, 1964) 508–509.

[15] G. S. Guralnik, C. R. Hagen and T. W. B. Kibble, *Global conservation laws and massless particles*, *Phys. Rev. Lett.* **13** (Nov, 1964) 585–587.

[16] N. Cabibbo, *Unitary symmetry and leptonic decays*, *Phys. Rev. Lett.* **10** (Jun, 1963) 531–533.

[17] M. Kobayashi and T. Maskawa, *CP-Violation in the Renormalizable Theory of Weak Interaction*, *Progress of Theoretical Physics* **49** (02, 1973) 652–657, [`https://academic.oup.com/ptp/article-pdf/49/2/652/5257692/49-2-652.pdf`].

[18] A. Hocker, H. Lacker, S. Laplace and F. Le Diberder, *A New approach to a global fit of the CKM matrix*, *Eur. Phys. J. C* **21** (2001) 225–259, [`hep-ph/0104062`].

[19] J. Charles, A. Höcker, H. Lacker, S. Laplace, F. R. Le Diberder, J. Malclés et al., *Cp violation and the ckm matrix: assessing the impact of the asymmetric b factories*, *The European Physical Journal C* **41** (May, 2005) 1–131.

[20] L.-L. Chau and W.-Y. Keung, *Comments on the parametrization of the kobayashi-maskawa matrix*, *Phys. Rev. Lett.* **53** (Nov, 1984) 1802–1805.

[21] L. Wolfenstein, *Parametrization of the kobayashi-maskawa matrix*, *Phys. Rev. Lett.* **51** (Nov, 1983) 1945–1947.

[22] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, *Journal of High Energy Physics* **2017** (Jan, 2017) .

[23] ATLAS Collaboration collaboration, *Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector*, tech. rep., CERN, Geneva, Jul, 2017.

[24] J. S. H. Lee, I. Park, I. J. Watson and S. Yang, *Quark-gluon jet discrimination using convolutional neural networks*, *Journal of the Korean Physical Society* **74** (Feb, 2019) 219–223.

[25] S. Badger and J. Bullock, *Using neural networks for efficient evaluation of high multiplicity scattering amplitudes*, *JHEP* **06** (2020) 114, [2002.07516].

[26] J. Aylett-Bullock, S. Badger and R. Moodie, *Optimising simulations for diphoton production at hadron colliders using amplitude neural networks*, 2106.09474.

[27] M. Feickert and B. Nachman, *A living review of machine learning for particle physics*, 2021.

[28] D. Guest, K. Cranmer and D. Whiteson, *Deep learning and its application to lhc physics*, *Annual Review of Nuclear and Particle Science* **68** (Oct, 2018) 161–181.

[29] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel et al., *Machine learning at the energy and intensity frontiers of particle physics*, *Nature* **560** (2018) 41–48.

[30] D. Bourilkov, *Machine and deep learning applications in particle physics*, *International Journal of Modern Physics A* **34** (Dec, 2019) 1930019.

[31] N. J. Nilsson, *Introduction to machine learning. an early draft of a proposed textbook*, 1996.

[32] H. Drucker and C. Cortes, *Boosting decision trees*, in *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, (Cambridge, MA, USA), p. 479–485, MIT Press, 1995.

[33] J. R. Quinlan, *Induction of decision trees*, *Machine Learning* **1** (1986) 81–106.

[34] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA, 1984.

[35] A. D'Ambrosio and V. A. Tutore, *Conditional classification trees by weighting the gini impurity measure*, in *New Perspectives in Statistical Modeling and Data Analysis* (S. Ingrassia, R. Rocci and M. Vichi, eds.), (Berlin, Heidelberg), pp. 273–280, Springer Berlin Heidelberg, 2011.

[36] Y.-Y. Song and Y. Lu, *Decision tree methods: applications for classification and prediction*, *Shanghai archives of psychiatry* **27** (04, 2015) 130–135.

[37] L. Breiman, *Random forests*, *Machine Learning* **45** (2001) 5–32.

[38] J. H. Friedman, *Greedy function approximation: a gradient boosting machine*, *Annals of statistics* (2001) 1189–1232.

[39] J. H. Friedman, *Stochastic gradient boosting*, *Comput. Stat. Data Anal.* **38** (Feb., 2002) 367–378.

[40] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016, DOI.

[41] R. M. Neal, *Bayesian Learning for Neural Networks.* Springer-Verlag, Berlin, Heidelberg, 1996.

[42] T. Charnock, L. Perreault-Levasseur and F. Lanusse, *Bayesian neural networks*, 2020.

[43] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain.*, *Psychological Review* **65** (1958) 386–408.

[44] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA, USA, 1969.

[45] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017.

[46] N. Qian, *On the momentum term in gradient descent learning algorithms*, *Neural Networks* **12** (1999) 145–151.

[47] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017.

[48] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, *Mathematics of Control, Signals and Systems* **2** (1989) 303–314.

[49] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, (Madison, WI, USA), p. 807–814, Omnipress, 2010.

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

[51] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, *Nature* **521** (2015) 436–444.

[52] L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications.* CRC Press, Inc., USA, 1st ed., 1999.

[53] Y. Wen, P. Vicol, J. Ba, D. Tran and R. Grosse, *Flipout: Efficient pseudo-independent weight perturbations on mini-batches*, 2018.

[54] S. Bollweg, M. Haussmann, G. Kasieczka, M. Luchmann, T. Plehn and J. Thompson, *Deep-learning jets with uncertainties and more*, *SciPost Physics* **8** (Jan, 2020) .

[55] A. Graves, *Practical variational inference for neural networks*, in *Advances in Neural Information Processing Systems* (J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira and K. Q. Weinberger, eds.), vol. 24, pp. 2348–2356, Curran Associates, Inc., 2011.

[56] J. Gallicchio, J. Huth, M. Kagan, M. D. Schwartz, K. Black and B. Tweedie, *Multivariate discrimination and the Higgs + W/Z search*, *JHEP* **04** (2011) 069, [1010.3698].

[57] A. Biekötter, K. W. Kwok and B. D. Pecjak, *Potential and limitations of machine-learning approaches to inclusive $|v_{ub}|$ determinations*, 2021.

[58] I. I. Y. Bigi, M. A. Shifman, N. G. Uraltsev and A. I. Vainshtein, *On the motion of heavy quarks inside hadrons: Universal distributions and inclusive decays*, *Int. J. Mod. Phys. A* **9** (1994) 2467–2504, [hep-ph/9312359].

[59] J. Chay, H. Georgi and B. Grinstein, *Lepton energy distributions in heavy meson decays from QCD*, *Phys. Lett. B* **247** (1990) 399–405.

[60] I. I. Y. Bigi, N. G. Uraltsev and A. I. Vainshtein, *Nonperturbative corrections to inclusive beauty and charm decays: QCD versus phenomenological models*, *Phys. Lett. B* **293** (1992) 430–436, [hep-ph/9207214].

[61] B. Blok, L. Koyrakh, M. A. Shifman and A. I. Vainshtein, *Differential distributions in semileptonic decays of the heavy flavors in QCD*, *Phys. Rev. D* **49** (1994) 3356, [hep-ph/9307247].

[62] A. V. Manohar and M. B. Wise, *Inclusive semileptonic B and polarized Lambda(b) decays from QCD*, *Phys. Rev. D* **49** (1994) 1310–1329, [hep-ph/9308246].

[63] T. van Ritbergen, *The Second order QCD contribution to the semileptonic b —> u decay rate*, *Phys. Lett. B* **454** (1999) 353–358, [hep-ph/9903226].

[64] M. Brucherseifer, F. Caola and K. Melnikov, *On the $O(\alpha_s^2)$ corrections to $b \to X_u e\bar{\nu}$ inclusive decays*, *Phys. Lett. B* **721** (2013) 107–110, [1302.0444].

[65] B. Capdevila, P. Gambino and S. Nandi, *Perturbative corrections to power suppressed effects in $\bar{B} \to X_u \ell\nu$*, *JHEP* **04** (2021) 137, [2102.03343].

[66] M. Neubert, *QCD based interpretation of the lepton spectrum in inclusive anti-B —> X(u) lepton anti-neutrino decays*, *Phys. Rev. D* **49** (1994) 3392–3398, [hep-ph/9311325].

[67] M. Neubert, *Analysis of the photon spectrum in inclusive $b \to x_s\gamma$ decays*, *Physical Review D* **49** (May, 1994) 4623–4633.

[68] K. S. M. Lee and I. W. Stewart, *Factorization for power corrections to B —> X(s) gamma and B —> X(u) l anti-nu*, *Nucl. Phys. B* **721** (2005) 325–406, [hep-ph/0409045].

[69] S. W. Bosch, M. Neubert and G. Paz, *Subleading shape functions in inclusive B decays*, *JHEP* **11** (2004) 073, [hep-ph/0409115].

[70] M. Beneke, F. Campanario, T. Mannel and B. D. Pecjak, *Power corrections to anti-B —> X(u) l anti-nu (X(s) gamma) decay spectra in the 'shape-function' region*, *JHEP* **06** (2005) 071, [hep-ph/0411395].

[71] C. Greub, M. Neubert and B. D. Pecjak, *NNLO corrections to anti-B —> X(u) l anti-nu(l) and the determination of |V(ub)|*, *Eur. Phys. J. C* **65** (2010) 501–515, [0909.1609].

[72] F. De Fazio and M. Neubert, *B —> X(u) lepton anti-neutrino lepton decay distributions to order alpha(s)*, *JHEP* **06** (1999) 017, [hep-ph/9905351].

[73] B. O. Lange, M. Neubert and G. Paz, *Theory of charmless inclusive B decays and the extraction of V(ub)*, *Phys. Rev. D* **72** (2005) 073006, [hep-ph/0504071].

[74] P. Gambino, P. Giordano, G. Ossola and N. Uraltsev, *Inclusive semileptonic B decays and the determination of |V(ub)|*, *JHEP* **10** (2007) 058, [0707.2493].

[75] A. Bharucha, D. M. Straub and R. Zwicky, *$b \to vl^+l^-$ in the standard model from light-cone sum rules*, *Journal of High Energy Physics* **2016** (Aug, 2016) .

[76] Belle collaboration, H. Ha et al., *Measurement of the decay $B^0 \to \pi^- \ell^+ \nu$ and determination of $|V_{ub}|$*, *Phys. Rev. D* **83** (2011) 071101, [1012.0090].

[77] Belle collaboration, A. Sibidanov et al., *Study of Exclusive $B \to X_u \ell \nu$ Decays and Extraction of $\|V_{ub}\|$ using Full Reconstruction Tagging at the Belle Experiment*, *Phys. Rev. D* **88** (2013) 032005, [1306.2781].

[78] BaBar collaboration, P. del Amo Sanchez et al., *Study of $B \to \pi \ell \nu$ and $B \to \rho \ell \nu$ Decays and Determination of $|V_{ub}|$*, *Phys. Rev. D* **83** (2011) 032007, [1005.3288].

[79] BaBar collaboration, J. Lees et al., *Branching fraction and form-factor shape measurements of exclusive charmless semileptonic B decays, and determination of $|V_{ub}|$*, *Phys. Rev. D* **86** (2012) 092004, [1208.1253].

[80] Belle collaboration, P. Urquijo et al., *Measurement Of |V(ub)| From Inclusive Charmless Semileptonic B Decays*, *Phys. Rev. Lett.* **104** (2010) 021801, [0907.0379].

[81] BaBar collaboration, J. Lees et al., *Study of $\bar{B} \to X_u \ell \bar{\nu}$ decays in $B\bar{B}$ events tagged by a fully reconstructed B-meson decay and determination of $\|V_{ub}\|$*, *Phys. Rev. D* **86** (2012) 032004, [1112.0702].

[82] Belle collaboration, L. Cao et al., *Measurements of Partial Branching Fractions of Inclusive $B \to X_u \, \ell^+ \, \nu_\ell$ Decays with Hadronic Tagging*, 2102.00020.

[83] LHCb collaboration, R. Aaij et al., *Determination of the quark coupling strength $|V_{ub}|$ using baryonic decays*, *Nature Phys.* **11** (2015) 743–747, [1504.01568].

[84] Particle Data Group collaboration, M. Tanabashi et al., *Review of Particle Physics*, *Phys. Rev. D* **98** (2018) 030001.

[85] M. Gelb, *Search for the Rare Decay* $\mathrm{B}^+ \to \ell^+\nu_\ell\gamma$ *with the Full Event Interpretation at the Belle Experiment*, Ph.D. thesis, Karlsruhe Institute of Technology (KIT), 2018.

[86] U. Aglietti, F. Di Lodovico, G. Ferrera and G. Ricciardi, *|V(ub)| extraction using the Analytic Coupling model*, *Nucl. Phys. B Proc. Suppl.* **185** (2008) 33–38, [0809.4860].

[87] S. W. Bosch, B. O. Lange, M. Neubert and G. Paz, *Factorization and shape function effects in inclusive B meson decays*, *Nucl. Phys. B* **699** (2004) 335–386, [hep-ph/0402094].

[88] J. R. Andersen and E. Gardi, *Inclusive spectra in charmless semileptonic B decays by dressed gluon exponentiation*, *JHEP* **01** (2006) 097, [hep-ph/0509360].

[89] P. Baldi, P. Sadowski and D. Whiteson, *Searching for Exotic Particles in High-Energy Physics with Deep Learning*, *Nature Commun.* **5** (2014) 4308, [1402.4735].

[90] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban and D. Whiteson, *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev. D* **94** (2016) 112002, [1607.08633].

[91] D. Guest, K. Cranmer and D. Whiteson, *Deep Learning and its Application to LHC Physics*, *Ann. Rev. Nucl. Part. Sci.* **68** (2018) 161–181, [1806.11484].

[92] D. Lange, *The EvtGen particle decay simulation package*, *Nucl. Instrum. Meth. A* **462** (2001) 152–155.

[93] T. Gleisberg et al., *Event generation with SHERPA 1.1*, *JHEP* **02** (2009) 007, [0811.4622].

[94] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159–177, [1410.3012].

[95] T. Sjostrand, S. Mrenna and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852–867, [0710.3820].

[96] E. Barberio, B. van Eijk and Z. Was, *PHOTOS: A Universal Monte Carlo for QED radiative corrections in decays*, *Comput. Phys. Commun.* **66** (1991) 115–128.

[97] E. Barberio and Z. Was, *PHOTOS: A Universal Monte Carlo for QED radiative corrections. Version 2.0*, *Comput. Phys. Commun.* **79** (1994) 291–308.

[98] TASSO collaboration, M. Althoff et al., *A Detailed Study of Strange Particle Production in e^+e^- Annihilation at High-energy*, *Z. Phys. C* **27** (1985) 27.

[99] JADE collaboration, W. Bartel et al., *Charged Particle and Neutral Kaon Production in e+ e- Annihilation at PETRA*, *Z. Phys. C* **20** (1983) 187.

[100] P. Skands, S. Carrazza and J. Rojo, *Tuning PYTHIA 8.1: the Monash 2013 Tune*, *Eur. Phys. J. C* **74** (2014) 3024, [1404.5630].

[101] M. Cacciari, G. P. Salam and G. Soyez, *The Anti-k(t) jet clustering algorithm*, *JHEP* **04** (2008) 063, [0802.1189].

[102] D. E. Kaplan, K. Rehermann, M. D. Schwartz and B. Tweedie, *Top tagging: A method for identifying boosted hadronically decaying top quarks*, *Physical Review Letters* **101** (Oct, 2008) .

[103] T. Plehn, M. Spannowsky, M. Takeuchi and D. Zerwas, *Stop reconstruction with tagged tops*, *Journal of High Energy Physics* **2010** (Oct, 2010) .

[104] T. Plehn, G. P. Salam and M. Spannowsky, *Fat jets for a light higgs boson*, *Physical Review Letters* **104** (Mar, 2010) .

[105] J. Thaler and K. Van Tilburg, *Identifying boosted objects with n-subjettiness*, *Journal of High Energy Physics* **2011** (Mar, 2011) .

[106] A. J. Larkoski, S. Marzani, G. Soyez and J. Thaler, *Soft drop*, *Journal of High Energy Physics* **2014** (May, 2014) .

[107] A. Strandlie and R. Frühwirth, *Track and vertex reconstruction: From classical to adaptive methods*, *Rev. Mod. Phys.* **82** (May, 2010) 1419–1458.

[108] ATLAS collaboration, S. Heer, *The secondary vertex finding algorithm with the ATLAS detector*, *PoS* **EPS-HEP2017** (2017) 762.

[109] A. Sirunyan, A. Tumasyan, W. Adam, F. Ambrogi, E. Asilar, T. Bergauer et al., *Identification of heavy-flavour jets with the cms detector in pp collisions at 13 tev*, *Journal of Instrumentation* **13** (May, 2018) P05011–P05011.

[110] J. Gallicchio and M. D. Schwartz, *Quark and gluon jet substructure*, *Journal of High Energy Physics* **2013** (Apr, 2013) .

[111] R. Field and R. Feynman, *A parametrization of the properties of quark jets*, *Nuclear Physics B* **136** (1978) 1–76.

[112] D. Krohn, M. D. Schwartz, T. Lin and W. J. Waalewijn, *Jet charge at the lhc*, *Physical Review Letters* **110** (May, 2013) .

[113] W. J. Waalewijn, *Calculating the charge of a jet*, *Physical Review D* **86** (Nov, 2012) .

[114] K. Fraser and M. D. Schwartz, *Jet charge and machine learning*, *Journal of High Energy Physics* **2018** (Oct, 2018) .

[115] *Jet Charge Studies with the ATLAS Detector Using $\sqrt{s} = 8$ TeV Proton-Proton Collision Data*, tech. rep., CERN, Geneva, Aug, 2013.

[116] B. Nachman, *Jet charge with the atlas detector using $\sqrt{s} = 8$ tev pp collision data*, 2014.

[117] ATLAS Collaboration collaboration, G. Aad, B. Abbott, J. Abdallah, O. Abdinov, R. Aben, M. Abolins et al., *Measurement of jet charge in dijet events from $\sqrt{s} = 8$* TeV *pp collisions with the atlas detector*, *Phys. Rev. D* **93** (Mar, 2016) 052003.

[118] G. Aad, B. Abbott, J. Abdallah, O. Abdinov, R. Aben, M. Abolins et al., *Measurement of jet charge in dijet events from $\sqrt{s} = 8$ tev pp collisions with the atlas detector*, *Physical Review D* **93** (Mar, 2016) .

[119] A. M. Sirunyan, A. Tumasyan, W. Adam, E. Asilar, T. Bergauer, J. Brandstetter et al., *Measurements of jet charge with dijet events in pp collisions at $\sqrt{s} = 8$ tev*, *Journal of High Energy Physics* **2017** (Oct, 2017) .

[120] ATLAS, CMS collaboration, S. Tokar, *Jet charge determination at the LHC*, in *Parton radiation and fragmentation from LHC to FCC-ee*, pp. 79–84, 2, 2017.

[121] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban and D. Whiteson, *Jet flavor classification in high-energy physics with deep neural networks*, *Physical Review D* **94** (Dec, 2016) .

[122] J. Shlomi, S. Ganguly, E. Gross, K. Cranmer, Y. Lipman, H. Serviansky et al., *Secondary vertex finding in jets with neural networks*, *The European Physical Journal C* **81** (Jun, 2021) .

[123] A. Ali, F. Barreiro and T. Lagouri, *Prospects of measuring the ckm matrix element $|V_{ts}|$ at the lhc*, *Physics Letters B* **693** (Sep, 2010) 44–51.

[124] J. Duarte-Campderros, G. Perez, M. Schlaffer and A. Soffer, *Probing the higgs–strange-quark coupling at e + e - colliders using light-jet flavor tagging*, *Physical Review D* **101** (06, 2020) .

[125] J. Erdmann, *A tagger for strange jets based on tracking information using long short-term memory*, *Journal of Instrumentation* **15** (Jan, 2020) P01021–P01021.

[126] J. Erdmann, O. Nackenhorst and S. Zeißner, *Maximum performance of strange-jet tagging at hadron colliders*, *Journal of Instrumentation* **16** (Aug, 2021) P08039.

[127] Y. Nakai, D. Shih and S. Thomas, *Strange jet tagging*, 2020.

[128] A. Powell, *Particle Identification at LHCb. Particle ID in LHCb*, .

[129] C. CMS, *A MIP Timing Detector for the CMS Phase-2 Upgrade*, Tech. Rep. CERN-LHCC-2019-003, CMS-TDR-020, CERN, Geneva, Mar, 2019.

[130] ATLAS Collaboration collaboration, *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade*, Tech. Rep. CERN-LHCC-2020-007, ATLAS-TDR-031, CERN, Geneva, Jun, 2020.

[131] F. Krauss, R. Kuhn and G. Soff, *AMEGIC++ 1.0: A Matrix element generator in C++*, *JHEP* **02** (2002) 044, [`hep-ph/0109036`].

[132] T. Gleisberg and S. Höche, *Comix, a new matrix element generator*, *Journal of High Energy Physics* **2008** (Dec, 2008) 039–039.

[133] S. Schumann and F. Krauss, *A Parton shower algorithm based on Catani-Seymour dipole factorisation*, *JHEP* **03** (2008) 038, [0709.1027].

[134] R. D. Ball, V. Bertone, S. Carrazza, C. S. Deans, L. Del Debbio, S. Forte et al., *Parton distributions for the lhc run ii*, *Journal of High Energy Physics* **2015** (Apr, 2015) .

[135] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht et al., *Lhapdf6: parton density access in the lhc precision era*, *The European Physical Journal C* **75** (Mar, 2015) .

[136] R. Aaij, C. Abellán Beteta, B. Adeva, M. Adinolfi, C. Aidala, Z. Ajaltouni et al., *Measurement of charged hadron production in z-tagged jets in proton-proton collisions at $\sqrt{s}$=8TeV*, *Physical Review Letters* **123** (Dec, 2019) .

[137] E. Conte, B. Fuks and G. Serret, *MadAnalysis 5, A User-Friendly Framework for Collider Phenomenology*, *Comput. Phys. Commun.* **184** (2013) 222–256, [1206.1599].

[138] E. Conte, B. Dumont, B. Fuks and C. Wymant, *Designing and recasting LHC analyses with MadAnalysis 5*, *Eur. Phys. J. C* **74** (2014) 3103, [1405.3982].

[139] J. Y. Araz, B. Fuks and G. Polykratis, *Simplified fast detector simulation in MADANALYSIS 5*, *Eur. Phys. J. C* **81** (2021) 329, [2006.09387].

[140] M. Cacciari, G. P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J.* **C72** (2012) 1896, [1111.6097].

[141] J. Duarte-Campderros, G. Perez, M. Schlaffer and A. Soffer, *Probing the Higgs–strange-quark coupling at e^+e^- colliders using light-jet flavor tagging*, *Phys. Rev. D* **101** (2020) 115005, [1811.09636].

[142] P. D. Group, P. A. Zyla, R. M. Barnett, J. Beringer, O. Dahl, D. A. Dwyer et al., *Review of Particle Physics*, *Progress of Theoretical and Experimental*

Physics **2020** (08, 2020) ,
[https://academic.oup.com/ptep/article-pdf/2020/8/083C01/34673722/ptaa104.

[143] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., "TensorFlow: Large-scale machine learning on heterogeneous systems." https://www.tensorflow.org/, 2015.

[144] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean et al., *Tensorflow: A system for large-scale machine learning*, *CoRR* **abs/1605.08695** (2016) , [1605.08695].

[145] F. Chollet et al., "Keras." https://keras.io, 2015.

[146] J. Bergstra, D. Yamins and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms."

[147] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[148] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et al., eds.), pp. 4765–4774. Curran Associates, Inc., 2017.

[149] A. Datta, S. Sen and Y. Zick, *Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems*, in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 598–617, 2016, DOI.

[150] S. Lipovetsky and M. Conklin, *Analysis of regression in game theory approach*, *Applied Stochastic Models in Business and Industry* **17** (2001) 319–330, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.446].

[151] E. Štrumbelj and I. Kononenko, *Explaining prediction models and individual predictions with feature contributions*, *Knowledge and Information Systems* **41** (2014) 647–665.

[152] LHCb Collaboration collaboration, R. Aaij, B. Adeva, M. Adinolfi, Z. Ajaltouni, S. Akar, J. Albrecht et al., *Expression of Interest for a Phase-II LHCb Upgrade: Opportunities in flavour physics, and beyond, in the HL-LHC era*, Tech. Rep. CERN-LHCC-2017-003, CERN, Geneva, Feb, 2017.

[153] L. collaboration, I. Bediaga, M. C. Torres, J. M. D. Miranda, A. Gomes, A. Massafferri et al., *Physics case for an lhcb upgrade ii - opportunities in flavour physics, and beyond, in the hl-lhc era*, 2019.

[154] E. Sinclair, *Option Trading: Pricing and Volatility Strategies and Techniques*. Wiley trading series. Wiley, 2010.

[155] R. F. Engle, *Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation*, *Econometrica* **50** (1982) 987–1007.

[156] T. Bollerslev, *Generalized autoregressive conditional heteroskedasticity*, *Journal of Econometrics* **31** (1986) 307 – 327.

[157] S. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, *Review of Financial Studies* **6** (1993) 327–343.

[158] L. Bauwens, C. Hafner and S. Laurent, *Handbook of volatility models and their applications*, *Handbook of Volatility Models and Their Applications* (03, 2012) .

[159] R. Engle and G. Lee, *A long-run and short-run component model of stock return volatility*, .

[160] L. P. Hansen, *Large sample properties of generalized method of moments estimators*, *Econometrica* **50** (1982) 1029–1054.

[161] A. van Haastrecht and A. Pelsser, *Efficient, almost exact simulation of the heston stochastic volatility model*, *International Journal of Theoretical and Applied Finance* **13** (2010) 1–43.

[162] E. Ramos-Pérez, P. J. Alonso-González and J. J. Núñez-Velázquez, *Forecasting volatility with a stacked model based on a hybridized artificial neural network*, *Expert Systems with Applications* **129** (Sep, 2019) 1–9.

[163] Y. L. Zhou, R. J. Han, Q. Xu, Q. J. Jiang and W. K. Zhang, *Long short-term memory networks for CSI300 volatility prediction with Baidu search volume*, *Concurrency Computation* **31** (2019) 1–7, [1805.11954].

[164] F. Black and M. Scholes, *The valuation of option contracts and a test of market efficiency*, *The Journal of Finance* **27** (1972) 399–417.

[165] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, *Journal of Political Economy* **81** (1973) 637–654.

[166] R. C. Merton, *Theory of rational option pricing*, *The Bell Journal of Economics and Management Science* **4** (1973) 141–183.

[167] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore et al., *Tensorflow distributions*, *CoRR* **abs/1711.10604** (2017) , [1711.10604].

[168] V. Cerqueira, L. Torgo, M. Oliveira and B. Pfahringer, *Dynamic and heterogeneous ensembles for time series forecasting*, pp. 242–251, 10, 2017, DOI.

[169] BaBar collaboration, B. Aubert et al., *The BABAR Detector: Upgrades, Operation and Performance*, *Nucl. Instrum. Meth. A* **729** (2013) 615–701, [1305.3560].

[170] N. Gagliardi, *Measurements of Partial Branching Fractions for Charmless Semileptonic B Decays with the BaBar Experiment and Determination of V_{ub}*, Ph.D. thesis, Università di Padova, 2009.

[171] BaBar collaboration, B. Aubert et al., *The First year of the BaBar experiment at PEP-II*, in *30th International Conference on High-Energy Physics*, 12, 2000, hep-ex/0012042.

[172] A. Buckley, D. Kar and K. Nordström, *Fast simulation of detector effects in Rivet*, *SciPost Phys.* **8** (2020) 025, [1910.01637].

[173] LHCb VELO Group collaboration, A. Oblakowska-Mucha, *The LHCb Vertex Locator - performance and radiation damage*, *JINST* **9** (2014) C01065.

[174] F. Archilli et al., *Performance of the Muon Identification at LHCb*, *JINST* **8** (2013) P10020, [1306.0249].

[175] DELPHES 3 collaboration, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens et al., *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057, [1307.6346].